QTERM®-IV/P40
USER'S MANUAL
REVISION 10

BEIJER ELECTRONICS
2212 South West Temple #50
Salt Lake City, Utah 84115-2648
USA
Phone 801-466-8770
Fax 801-466-8792
Email info@beijerelectronicsinc.com
Web http://www.beijerelectronicsinc.com

M01-008-00 Rev 10

9241E2 - Printed in USA

© Copyright Beijer Electronics 2011

QTERM, QTERM-IV, QTERM-P40, QCODE, QDATA, QSETUP and QDEMO are trademarks of Beijer Electronics.

Manual Updated 21 March 2011

FCC COMPLIANCE STATEMENT

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Any modification to this device (including any changes to the recommended antenna configuration) that are not expressly approved by Beijer Electronics could void the user's authority to operate this device.

FOREWORD

The Beijer Electronics QTERM® is a data-entry terminal for industrial applications. There are two distinct QTERM models covered in this manual; the QTERM-IV and the QTERM-P40. When the term QTERM is used in this manual, it refers to any one of these models. When the manual is referring to one of these specific models, then it will be indicated. The QTERM is available in many different configurations; this manual discusses all versions and their operation.

The notation ₵ in this manual always means the one-byte escape character (1Bh = 27 decimal).

This manual is written for Version 2.1 of the QTERM operation code. Be sure to read the file READ.ME on your distribution disk for any changes due to newer versions of code.

The QTERM-IV and QTERM-P40 are CE certified products. They have been assessed against the requirements of EN 50082-1: 1992, EN 55022: 1987, and EN 60950 (including Amendments Nos. 1, 2, and 3). Based on conformity with these requirements, the QTERM-IV and QTERM-P40 are deemed in compliance with all applicable CE directives.

The sections of this manual are as follows:

Chapter 1 **Quick Start.** If you wish to start using your QTERM immediately, this chapter will show you how to connect to the terminal and how to run the example software.

Chapter 2 **QTERM Software.** This chapter provides a detailed listing of the QTERM commands and discusses how to use them.

Chapter 3 **Using The QDATA File.** The QDATA file is used to configure the QTERM. This chapter shows you how to modify the file and load it into your QTERM.

Chapter 4 **QTERM Serial Modes.** This chapter discusses the different communication modes available including full-duplex, block serial and multidrop serial modes.

Chapter 5 **QTERM-IV Hardware.** This chapter discusses the hardware of the QTERM-IV, including dimensional drawings, interface specifications, connector pin assignments, performance and environmental specifications.

Chapter 6 **QTERM-P40 Hardware.** This chapter discusses the hardware of the QTERM-P40, including dimensional drawings, interface specifications, connector pin assignments, performance and environmental specifications.

Appendix A **ASCII Chart.** This is a true 7-bit ASCII chart, along with mnemonic definitions.

Appendix B **QTERM Character Chart.** This is a 256-character chart showing how the QTERM handles every character it receives. The lower half is similar to, but not the same as, the true ASCII chart in Appendix A.

Appendix C **QTERM Command Summary.** This is an abbreviated summary of QTERM software commands.

CHAPTER 1.

QUICK START

There are only four steps required to communicate with the QTERM:

- use *Power-On Setup* to set the contrast, baud rate, data format and serial mode
- connect to your host transmit, receive and ground lines
- apply power
- transmit and receive with the QTERM

1.1 Power-On Setup

The *Power-On Setup* procedure is used to configure the QTERM's display contrast, baud rate, data format and serial mode. You use three different QTERM keys to do this configuration (see Figure 1-1):

For the QTERM-IV:

Top - Right: this is the *up* key
Top - 2nd from Right: this is the *down* key
Top - 3rd from Right: this is the *save* key

For the QTERM-P40:

Left - Top: this is the *up* key
Left - 2nd from Top: this is the *down* key
Left - 3rd from Top: this is the *save* key

To perform the *Power-On Setup* follow these steps:

- Hold down any key and apply power to the QTERM (you do not need to connect the transmit and receive lines).
- The version of software in the QTERM will be displayed as long as you hold down the key, after which the you can adjust the display contrast.
- Set the desired display contrast using the *up* and *down* keys. When the display is at a contrast you like, press the *save* key.
- Set the desired baud rate (see Table 1-1) using the *up* and *down* keys. When the desired baud rate is displayed, press the *save* key.
- Set the desired data format (see Table 1-1) using the *up* and *down* keys. When the desired data format is displayed, press the *save* key.
- Set the desired serial mode (see Table 1-1) using the *up* and *down* keys. When the desired serial mode is displayed, press the save key.

Disconnect power and go to the next step.

1.2 Connect the Communications Lines

The EIA-232 and 5-volt Buffered have one transmit and one receive line, while the EIA-422 has two transmit and two receive lines.

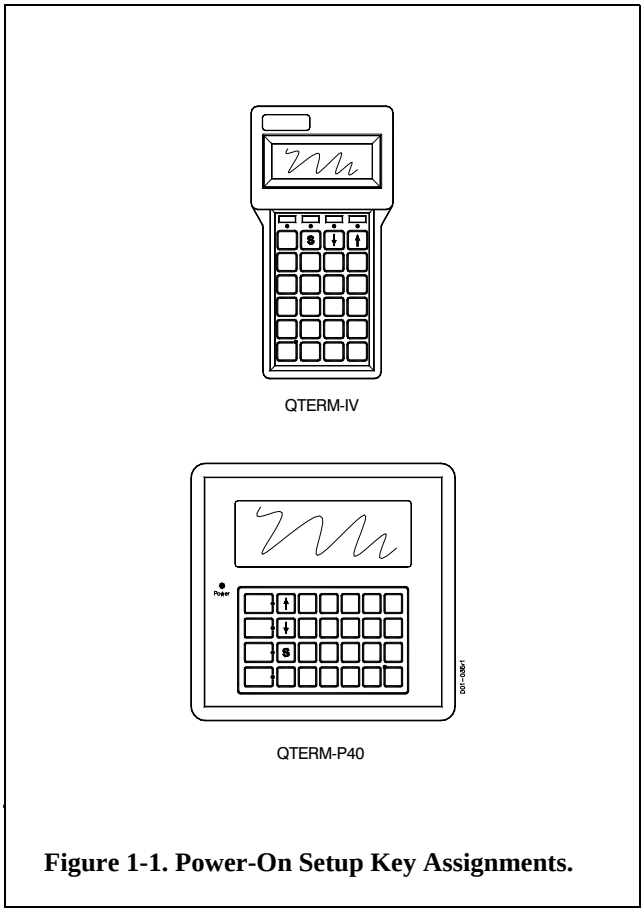


Figure 1-1. Power-On Setup Key Assignments.

Table 1-2 shows the connector pin assignments for both the handheld and panel-mount QTERMs. The receive and transmit directions shown in the table are with respect to the QTERM. Refer to this table to connect your host communications lines to the correct pins. Do not connect to the *dig0* and *dig1* pins; Chapter 2 discusses these pins and their uses.

(If you are using an IBM-style PC, you cannot connect an EIA-422 or 5-volt Buffered QTERM directly to the computer's COM port; you must provide an interface device.)

1.3 Apply Power

Table 1-2 shows the pin assignments for the power and ground lines. Connect your DC power supply to the appropriate two pins.

WARNING: Power supplied to the QTERM-IV/P40 must be from an SELV power source, and should have a current limit on its output of 5 Amperes. If you did not order the regulator option, the supply must provide a minimum of 4.75 volts DC and be limited to a maximum of 5.25 volts DC. If you did order the regulator option, the supply must provide a minimum of 5.5 volts DC and be limited to a maximum of 24 volts DC. Limiting may be inherent to the supply, or may be provided by supplementary overcurrent devices. If the QTERM does not respond, or exhibits abnormal behavior on power up, disconnect power and contact Beijer Electronics for technical support. If you have a QTERM-P40, then you can switch the regulator in or out; see Chapter 6 for information on setting the dip switches.

If you ordered the battery back option, you do not need to connect a power supply; just install six alkaline AA batteries in the battery compartment.

1.4 Communicate

At this point, characters which are transmitted by the host will be displayed on the QTERM display. If you press keys on the keypad, the QTERM will transmit the appropriate codes to the host computer.

1.5 Running QDEMO

To easily use some of the advanced features of the QTERM, run the QDEMO program that is included with this User's Manual:

Table 1-1: Power-on Setup Options.

Available Baud Rates
9600, 4800, 2400, 1200, 600, 300, 150
Available Data Formats*
8n1, 8e1, 8o1, 8n2, 8e2, 8o2, 7n1, 7e1, 7o1, 7n2, 7e2, 7o2
Available Serial Modes
fdplx = Full Duplex, Block, mdrop = Multidrop
Available Addresses for Multidrop Mode
Between 01 and 99
*format is: data-parity stop

- Connect your QTERM to either the COM1 or COM2 serial port on an IBM-style PC.

  (If you have an EIA-422 or 5-volt Buffered QTERM, you will need to provide an interface unit such as Beijer Electronics' QCOM-2; *you cannot directly connect a QTERM with one of these interfaces to the COM port on an IBM-style computer!)*

- Copy the file QDEMO.EXE to your hard disk.
- Apply power to the QTERM.
- Run the QDEMO program.
- Follow the on-screen instructions to use many of the features of the QTERM terminal.

QDEMO will allow you to exercise many of the features of the QTERM. It includes a terminal emulator so that you can also communicate directly with the terminal.

1.6 Using the QTERM

There are three ways to "customize" the operation of the QTERM for your application:

- Power-On Setup

  To set the display contrast, baud rate, data format and serial mode, see the beginning of this chapter.

- Software Commands

  If you want to send commands from your host computer to the QTERM, see Chapter 2 for a list of all commands and details on how to use them.

Table 1-2: QTERM Pin Assignments.

6-pin Modular	DB9 Female	Panel-mount 2 mm	10-pin Terminal Strip	QTERM 232/5-volt Buffered Function	QTERM 422 Function	QTERM 485 Function
1	3	1	4	receive	+receive	dig0*
2	6	2	5	dig1*	-receive	dig1*
3	2	3	6	transmit	+transmit	RTx+
4	1	4	7	dig0*	-transmit	RTx-
5	9	5	8	+V†	+V†	+V†
6	5	6	9	ground	ground	ground
-	n/a	-	1	E-stop N.O.	E-stop N.O.	E-stop N.O.
-	8	-	2	E-stop com.	E-stop com.	E-stop com.
-	7	-	3	E-stop N.C.	E-stop N.C.	E-stop N.C.

Tx and *Rx* directions are with respect to the QTERM
*these lines are optional; see section 5.5.3 and section 6.3.3
†no connection with battery back units

Some of the commands allow you to control the display (such as cursor movement, clear screen and line feed), others control operation of the hardware (such as LED control and key click control) and others control external devices (Auxiliary Serial Port control and digital output control).

- QDATA File Download

All configurable parameters of the QTERM can be controlled using software commands or by setting the proper values in the *QDATA file* and downloading it into the QTERM.

The *only* way to modify the key strings (i.e. what the QTERM transmits when a key is pressed), or to modify the eight custom characters, is by modifying and downloading the QDATA file.

Chapter 3 shows how to modify this file and how to load it into your QTERM. There are also several examples that utilize the powerful capabilities of key strings.

CHAPTER 2.

QTERM SOFTWARE

2.1 Operation

The operation of the QTERM is quite simple:

- Power is applied to the QTERM.
- Commands or data can be transmitted to the QTERM. Commands are executed as required; data is displayed.
- When a key is pressed, the QTERM transmits the appropriate character(s) to the host.

The QTERM has many additional capabilities which are accessed in three ways:

- Power-On Setup (see Chapter 1)
- Software commands from host (this chapter)
- Configuration-file download (see Chapter 3)

2.1.1 Handshaking

The QTERM has buffers for both receiving and transmitting characters. However, these buffers may not be large enough for some applications. If this is the case for your application, then your host must use XON/XOFF handshaking to make optimal use of the QTERM's capabilities.

The QTERM's transmit buffer (16 bytes long) is used when a host does not wish to receive characters for a period of time. The host sends an XOFF character to the QTERM. The QTERM starts placing keycodes (one character per keystroke, regardless of how many characters are in the string assigned to the key) into the transmit buffer rather than transmitting them. When the host is ready to receive characters, it sends an XON character, at which time the QTERM will start transmitting the characters in the buffer.

The QTERM's receive buffer is large enough to accept a full screen of text at the maximum baud rate (9600) without overrunning the buffer. However, some operations can take a longer time to execute, such as scrolling the entire screen down one line. In this case, the QTERM will send an XOFF to the host when it has only six bytes empty in its buffer. When it has processed input data to the point that the buffer only has six bytes used, it will send an XON and the host can resume transmitting.

The command summary in Appendix C shows approximate execution time for each of the various operations of the QTERM.

XON/XOFF handshaking can also be disabled (by software command or the QDATA file) if you do not wish to use it. However, if you do not use handshaking, you must take extra care that you do not overrun the QTERM's receive buffer. The two easiest ways to do this are to use a slower baud rate (which gives the QTERM more time to process each byte), or add delays to your code after each write to the QTERM.

2.1.2 Commands vs. QDATA Parameters

There are numerous parameters which are set by the QDATA configuration file (see Chapter 3), but which can be overridden by a software command. For example, tab spacing is set in the QDATA file but can be changed by the **Set Tab Spacing** command.

In all such cases, the software command overrides the QDATA configuration file setting but does not store the new setting into EEPROM. Therefore, when power is turned off, the overriding value is lost, and when the terminal is turned on, the QDATA parameter will be used again.

If you wish to have the software command permanently override the QDATA configuration file setting, you can use the **Store Parameters to EEPROM** command. This saves all currently-set parameters to the EEPROM which makes them the current power-on defaults.

2.1.3 User Data Memory Organization

The QTERM has about 22 Kbytes of storage for user data. Some of this is used for the information from the QDATA file, so the exact amount depends on how much data you put into key strings, macro strings. Typically, you will have about 20 to 21 kbytes available for user data.

Table 2-1: QTERM Software Commands.

Command	Code	Command	Code
Bell (^G)	07h	Auto Wrap Mode	⋐ R..
Backspace	08h	Auto Scroll Mode	⋐ S..
Horizontal Tab (^I)	09h	Auto Line Feed Mode	⋐ T..
Line Feed (^J)	0Ah	Display Off	⋐ U
Vertical Tab (^K)	0Bh	Backlight On/Off/Toggle	⋐ V..
Form Feed (^L)	0Ch	Query Status	⋐ W
Carriage Return (^M)	0Dh	Query Cursor Position	⋐ X
XON (DC1 or ^Q)	11h	Query Character	⋐ Y
XOFF (DC3 or ^S)	13h	Scroll Down	⋐ Z
Delete	7Fh	Key Repeat/Click Mode	⋐ a..
Cursor Up	⋐ A	Set Cursor Mode	⋐ b..
Cursor Down	⋐ B	Set Shift Mode	⋐ c..
Cursor Right	⋐ C	Control *dig0* and *dig1*	⋐ d..
Cursor Left	⋐ D	Set Buzzer Duration	⋐ e..
Clear Screen	⋐ E	Set Key Click Duration	⋐ f..
Set Timeout Delay	⋐ F..	Set Key Repeat Rate	⋐ g..
Set Tab Spacing	⋐ G..	Set LED Blink Rate	⋐ h..
Cursor Home	⋐ H	Save Parameters to EEPROM	⋐ i
Set Cursor Position	⋐ I..	Auxiliary Serial Port Control	⋐ j..
Erase to End of Screen	⋐ J	Transmit Buffer Flush	⋐ k
Erase to End of Line	⋐ K	XON/XOFF Mode	⋐ l..
Set COntrast	⋐ L..	User Area Read/Write	⋐ m..
Reset QTERM	⋐ M	Get Free User Area	⋐ n
Query Version	⋐ N	User Area Page Control	⋐ o..
Buzzer On/Off/Beep	⋐ O..	Macro String Execution	⋐ p..
LEDs On/Off/Blink/Toggle	⋐ P..	Query Multidrop Buffer	⋐ q
Insert/Replace Mode	⋐ Q..		

The user data memory is divided into 64-byte pages, of which one byte is always used for an “amount used” counter, leaving 63 bytes for actual data in each page. There will be around 330 pages available for user data with a typical-size QDATA file.

Pages are addressed using a “base 64” technique that guarantees that all addresses are valid ASCII characters. Each page address consists of two characters, ranging from ASCII “@” (40h) to DEL (7Fh). For example, “AA”, “3v”, “n]” are valid page addresses; “(F” is invalid because “(” is not in the 40h to 7Fh range.

Here are the formulas for converting between ASCII page numbers (MSB and LSB) and decimal page numbers (Page-Num):

Decimal to ASCII page conversion:

MSB = “@” + INT(Page-Num / 64)
LSB = “@” + (Page-Num MOD 64)

ASCII to Decimal page conversion:

Page-Num = (MSB - “@”) * 64 + (LSB - “@”)

All user area commands which do not explicitly specify a page use the current (i.e. last specified) data page.

Due to RAM limitations, multidrop terminals (EIA-485 and EIA-422) can only access the first 15 bytes of each page.

2.2 Software Commands

Once you have configured the QTERM using the power-on setup (see Chapter 1) and configuration-file download (see Chapter 3), you can program your host to control the QTERM using software commands.

Table 2-1 lists the commands available. Each command is discussed in detail below, and a command summary, including execution times, is given in Appendix C.

The notation ⋐ in this chapter always means the one-byte escape character (1Bh = 27 decimal).

Be sure to read the file READ.ME on your distribution disk for any late changes to this manual.

2.2.1 Bell (^G) - 07h

This causes the buzzer to beep. The default beep duration (½ second) is set in the QDATA file; it can be overridden with the *Set Buzzer Duration* command (ESC e). Note that sending this command is identical to sending the *Buzzer On/Off/Beep* command for a beep (ESC O B).

2.2.2 Backspace (^H) - 08h

Causes a nondestructive backspace, i.e. characters are not erased as the cursor is backspaced over them. With auto wrap mode off, the backspace stops at the left edge of the current display line. With auto wrap mode on, the cursor will wrap to the last position on the previous line. The command is ignored if the cursor is at the home position. See also the *Delete* command (7Fh).

2.2.3 Horizontal Tab (^I) - 09h

Moves the cursor right to the next tab column. The default tab spacing is set in the QDATA file; it can be overridden with the *Set Tab Spacing* command (ESC G). (The QDATA default tab spacing is every 4 spaces.)

With auto wrap on, the cursor will wrap down to the first column in the line below when it is tabbed beyond the last column in the current line. If auto wrap is off, the cursor will stop at the end of the current line. If auto scroll and auto wrap are both on, then the display will scroll up as the cursor is tabbed beyond the last column in the last line.

2.2.4 Line Feed (^J) - 0Ah

Moves the cursor down one line without changing its horizontal position. When auto scroll mode is on and a line feed is performed on the last line, the display will scroll up with the horizontal cursor position unaltered.

2.2.5 Vertical Tab (^K) - 0Bh

Performs the same function as *Line Feed*.

2.2.6 Form Feed (^L) - 0Ch

Performs the same function as *Line Feed*.

2.2.7 Carriage Return (^M) - 0Dh

Moves the cursor to left edge of the display on the current line. If auto line feed mode is on, then the cursor moves to the left edge of the next line. If auto scroll and auto line feed are both on, a carriage return on the last line will cause the display to scroll up and the cursor to be positioned at the left edge of the last line.

2.2.8 XON (^Q) - 11h

Enables the QTERM to transmit keys pushed after receiving an XOFF. XON is used to re-enable QTERM transmission after an XOFF has disabled it, allowing handshaking with the host system.

2.2.9 XOFF (^S) - 13h

Disables all QTERM transmission except for information requested via the *Query Status* command (ESC W). After receiving an XOFF command, the QTERM stores characters typed on the keypad in a transmit buffer. These characters will be transmitted when an XON is received. If the buffer becomes full before an XON is received, additional characters which are typed will be ignored.

2.2.10 Delete - 7FH

Delete works in the same way as *Backspace* (08h, ^H), except that characters are erased as the cursor moves over them.

2.2.11 Cursor Up - ESC A

Moves the cursor up one line without changing its horizontal position. It has no effect if the cursor is on the first line.

2.2.12 Cursor Down - ESC B

Moves the cursor down one line without changing its horizontal position. It has no effect if the cursor is on the last line.

2.2.13 Cursor Right - ESC C

Moves the cursor right one space without changing its vertical position. It has no effect if the cursor is at the right-most position on the current line.

2.2.14 Cursor Left - ESC D

Moves the cursor left one space without changing its vertical position. It has no effect if the cursor is at the left-most position on the current line.

2.2.15 Clear Screen - ESC E

Clears the display and moves the display cursor to home (the left-most position in the top line of the display).

2.2.16 Set Timeout Delay - ESC F

This sets the time after which, if no activity has occurred, the QTERM will automatically turn off the display (in

order to save power). Once the display has been turned off, it is turned back on by either of two events:

- If the operator presses a key, the display is turned back on. However, the key is *not* transmitted to the host.
- If the host sends a command or character to the QTERM, the display is turned back on, and the command or character will be processed normally.

If the QTERM has a backlight, and it is on when the timeout occurs, the backlight is turned off along with the display. When the display is turned back on, the backlight is turned on if it was on when the timeout occurred.

Anything displayed on the LCD screen when the timeout occurs is lost; i.e. when the display is turned back on it will be blank. Custom character definitions are *not* lost.

The format for this command is ⍋ *F #*, where # is a character in the range of 40h to 7Fh ("@" to DEL). 40h disables the timeout function. Other values set the timeout delay in 20 second increments, from 20 seconds through 21 minutes. See Table 1-2 (at the end of this chapter) for a complete list of codes and timeout delays.

The only thing the host can do that will not turn on the display is to issue the *Query Status* command (⍋ W).

2.2.17 Set Tab Spacing - ⍋ G

This sets the width of the horizontal tab columns. The command has the form ⍋ *G #*, where # is a character in the range of 40h to 53h ("@" to "S"). 40h causes tab characters to be ignored; 41h sets tabs at every column (which causes tabs to be treated as spaces), 42h sets tabs to every second column, etc. If # is outside this range, the command sequence is ignored.

For example, the three-byte command string

⍋ G D

sets tab spacing to every four spaces. See Table 1-2 (at the end of this chapter) for a complete list of codes and tab spacings.

2.2.18 Cursor Home - ⍋ H

Moves the cursor to the home (top left) position on the display.

2.2.19 Set Cursor Position - ⍋ I # *

Positions the cursor to the specified location (Table 1-2 shows location numbering). The command has the form *ESC I # **, where # sets the row and * sets the column. For example, the string

⍋ I B D

sets the cursor to row 2 (third row) and column 4 (fifth column). See Table 1-2 for a complete list of valid codes and cursor positions.

2.2.20 Erase to End of Screen - ⍋ J

Erases from the current cursor position to the end of the screen. The cursor position is unchanged.

2.2.21 Erase to End of Line - ⍋ K

Erases all displayed characters from the current cursor position to the end of the line. The cursor position is unchanged.

2.2.22 Set Contrast - ⍋ L

This command sets the display contrast. It has the form ⍋ *L #*, where # is in the range of 40h to 7Fh ("@" to DEL). The smaller the ASCII value of the character, the lower the contrast. The higher the ASCII value of the character, the higher the contrast.

2.2.23 Reset QTERM - ⍋ M

Resets the QTERM to its power-up state. This includes clearing all input and output buffers and the display, and resetting all parameters to the configuration set by the QDATA file, or a subsequent *Save Parameters to EEPROM* command.

2.2.24 Query Version - ⍋ N

This tells the QTERM to transmit its software version to the host. The version will consist of four ASCII characters in the format *vx.y*, where x and y are single ASCII digits.

2.2.25 Buzzer On/Off/Beep -⍋ O

This command controls the buzzer. It has the form ⍋ *O #*, where # is

@ -	turn buzzer off
A -	turn buzzer on
B -	beep for fixed duration

The fixed duration beep command (⍋ O B) is identical to sending a *Bell* command (07h, ^G). The duration used for beeping is set by the *Set Buzzer Duration* command (⍋ e) and by the QDATA file (default is ½ second).

2.2.26 LEDs On/Off/Blink/Toggle - ⋐ P

This controls the LEDs on the QTERM keypad, including the shift key LED. Any of these LEDs can be set to on, off or blink (the blink rate is set in the QDATA file and by the *Set LED Blink Rate* command). If a shift key is used, pressing the shift key will cause the proper state of the shift LED to override whatever state is set by this command. If no shift key is used, the shift LED is controlled solely by this command.

See Table 1-2 for a list of valid values for # and the effect of each such value. As an example of this command, the string

⋐ P R

sets the second LED from the right to blink.

2.2.27 Insert/Replace Mode - ⋐ Q

In insert mode, when a character is displayed at any position on the display, all following characters are moved right one position. The character in the current cursor location is also moved before the incoming character is displayed. In delete mode, the character displayed simply over writes whatever character was there previously.

Valid values for # are

@ - set replace mode
A - set insert mode

2.2.28 Auto Wrap Mode - ⋐ R

The auto wrap mode determines what happens when the cursor moves past the end of a line. With auto wrap off, the cursor stays at the last position in the line. With auto wrap on, the cursor moves down to the first position in the next line.

If the cursor moves past the end of the last line, and auto wrap is on, then the action depends on the auto scroll mode. If auto scroll is off, the cursor will wrap to the first position of the line, but the display will not scroll. Otherwise, the display will scroll, and the cursor will return to the first position in the last line.

Valid values for # are

@ - auto wrap off
A - auto wrap on

2.2.29 Auto Scroll Mode - ⋐ S

Auto scroll mode determines what happens when the cursor moves past the end of the last line. With auto scroll off, the cursor will stay in the last position. With auto scroll on, the display scrolls (i.e. all lines move up, and the last line becomes blank), and the cursor moves to the first position in the last line.

Valid values for # are

@ - auto scroll off
A - auto scroll on

2.2.30 Auto Line Feed Mode - ⋐ T

With auto line feed off, when a carriage return is received, the cursor returns to the first position in the current line.

With auto line feed on, the cursor moves to the first position in the next line, i.e. it acts as if both a carriage return and a line feed had been received.

Valid values for # are

@ - auto line feed off
A - auto line feed on

2.2.31 Display Off - ⋐ U

This turns off the display. Once the display has been turned off, it is turned back on by either of two events:

- If the operator presses a key, the display is turned back on. However, the key is *not* transmitted to the host.
- If the host sends a command or character to the QTERM, the display is turned back on, and the command or character will be processed normally.

If the QTERM has a backlight, and it is on when the command is received, the backlight is turned off along with the display. When the display is turned back on, the backlight is turned on if it was on when the command was processed.

Anything displayed on the LCD screen when the command is processed is lost; i.e. when the display is turned back on it will be blank. Custom character definitions are *not* lost.

The only thing the host can do that will not turn on the display is to issue the *Query Status* command (⋐ W).

The only reason to use this command is that it substantially reduces the power consumption of the QTERM.

2.2.32 Backlight On/Off/Toggle - ESC V

This command turns the backlight on and off.

Valid values for # are

@ - backlight off
A - backlight on
B - backlight toggle

2.2.33 Query Status - ESC W

The Query Status command (ESC W) returns a character indicating two things: the number of characters presently in the QTERM transmit buffer (0 to 15), and whether the display is on or off. Refer to Table 1-2 at the end of this chapter for the exact values returned for various conditions.

This is the only command which you can send to the QTERM which does not cause the display to be turned back on if the display is off when this command is received.

2.2.34 Query Cursor Position - ESC X

Returns the cursor position as two ASCII characters. These two characters are defined in the same way as for the *Set Cursor Position* (ESC I) command. See Table 1-2 for a detailed listing.

2.2.35 Query Character - ESC Y

Returns the character value for the character at the current cursor position.

2.2.36 Scroll Down - ESC Z

This causes the top three lines on the display to move down one line. The top line is cleared, and the cursor remains at the same absolute location on the display; i.e. it does not scroll down with the text.

2.2.37 Key Click/Repeat Mode - ESC a

This selectively enables and disables both key repeat and key click. The key click duration and key click rate are set in the QDATA file.

Valid values for # are

@ - click off, repeat off
A - click off, repeat on
B - click on, repeat off
C - click on, repeat on

This command globally enables and disables the key repeat. Which specific keys repeat is defined in the QDATA file.

2.2.38 Set Cursor Mode - ESC b

The QTERM cursor can be an underline cursor, a block cursor, neither or both. When neither is selected, no cursor is visible to the user.

Valid values for # are

@ - block off, underline off (no cursor)
A - block off, underline on
B - block on, underline off
C - block on, underline on

2.2.39 Set Shift Mode - ESC c

The shift key on the QTERM can operate in one of two ways:

- FUNCTION MODE, where the shift key stays shifted for one additional key press only
- LOCK MODE, where the shift key stays shifted until it is pressed a second time

For either mode, the shift LED can be enabled or disabled, and, if enabled, properly reflects the state of the shift key.

Valid values for # are

@ - function mode, shift LED enabled
A - lock mode, shift LED enabled
B - function mode, shift LED disabled
C - lock mode, shift LED disabled

Note that the shift LED enable/disable does not affect the shift operation itself, only the LED indicator. If you do not wish to have a shift key at all, you can disable the shift function by a parameter in the QDATA file.

2.2.40 Control *dig0* and *dig1* - ESC d

These two digital lines can be used for general control purposes, or they can be used to drive an external buzzer or horn.

Valid values for # are

@ - dig0 off (low)
A - dig1 off (low)
H - dig0 on (high)
I - dig1 on (high)
P - dig0 blink

Q - dig1 blink
X - dig0 buzzer
Y - dig1 buzzer
' - dig0 toggle
a - dig1 toggle

The blink mode blinks the line at the rate defined for LED blinking in the QDATA file. The buzzer mode causes the line to go on (high) when the internal buzzer is on, and to go off (low) when the internal buzzer is off. The toggle mode reverses the current state of the line. If the line is blinking and a toggle command is sent, the line will be toggled from its current state and will remain there (i.e., it will stop blinking).

2.2.41 Set Buzzer Duration - ⍷ e

This sets the duration for a "beep" command. The beep command can either be a *Bell* (07h, ^G) or the *Buzzer On/Off/Beep* command for beep (⍷ O B). The valid range for # is 41h through 7Fh ("A" to DEL). See Table 1-2 for the duration set by each character.

2.2.42 Set Key Click Duration - ⍷ f

This sets the length of the key click, which is actually a very short beep from the buzzer. The valid range for # is 41h through 7Fh ("A" to DEL). See Table 1-2 for the duration set by each character. Use the *Key Click/-Repeat Mode* command (⍷ a) to disable the key click.

2.2.43 Set Key Repeat Rate - ⍷ g

This sets the key repeat rate. The valid range for # is 41h through 7Fh ("A" to DEL). See Table 1-2 for the rate set by each character. See the *Key Click/Repeat Mode* command (⍷ a) to globally disable key repeats. Use the QDATA file to disable key repeats for selected keys.

2.2.44 Set LED Blink Rate - ⍷ h

This sets the LED blink rate and the PENDING mode LCD flash rate. The LEDs and the LCD flash at the same rate.. The valid range for # is 41h through 7Fh ("A" to DEL). See Table 1-2 for the rate set by each code. See also *LEDs On/Off/Blink* (⍷ P). See Chapter 5 for more information on PENDING mode.

2.2.45 Save Parameters to EEPROM - ⍷ i

This command causes all parameters to be stored to EEPROM. Any QDATA defaults which have been modified by software commands will be replaced by their modified values.

Note that the EEPROM used in the QTERM has a guaranteed write endurance of 10,000 write cycles. Although this is far more than most applications will ever require, if you use this command frequently in your application, you should take this endurance into consideration.

2.2.46 Auxiliary Serial Port Control - ⍷ j

This command is used to control the auxiliary serial port, if one is installed. The Auxiliary Port operation is mostly transparent to the QTERM: data sent to the Auxiliary Port is simply transmitted as received, and data received from theAuxiliary Port is transmitted directly to the host. The Auxiliary serial port does not support handshaking, which requires that the host interface run at 1200 baud or faster.

The are three valid values for #:

@ - disable receive from auxiliary port
A - enable receive from auxiliary port
B - transmit a string to the auxiliary port

DISABLE RECEIVE FROM AUXILIARY PORT: if # = "@", anything which the auxiliary port receives will be ignored.

ENABLE RECEIVE FROM AUXILIARY PORT: if # = "A", anything which is received by the auxiliary port will be transmitted to the host. Bytes are transmitted exactly as received, so the host will receive the data as if the device connected to the auxiliary port was transmitting directly to the host serial port.

TRANSMIT TO AUXILIARY PORT: if # = "B", then the host is transmitting data to the auxiliary port. The complete form of this command has this format:

⍷ j B * . . . .

where * indicates how many bytes are to follow in the ". . . ." string.

The first four characters (⍷ j B *) are *not* transmitted to the auxiliary port; the ". . . ." string is transmitted without any modification.

The character * must have a value of 41h through 7Fh ("A" to DEL), which indicates that the string will be from 1 byte to 63 bytes long. See Table 1-2 for a listing of values for * and the corresponding string length.

As an example of using this QTERM command, consider the situation where a Hewlett-Packard SmartWand barcode reader is connected to the Auxiliary Port. To send a

command string to the bar code wand, assume we wish to send this string to the wand:

ESC - y 4 Y F A I L

(This command would tell a Hewlett-Packard SmartWand to send the string "FAIL" if a read failure occurred.) This string is nine characters long, so the exact string to transmit to the QTERM would be

ESC j B I ESC - y 4 Y F A I L

The first four characters tell the QTERM to transmit the following nine characters directly to the bar code wand.

2.2.47 Transmit Buffer Flush - ESC k

If the host has transmitted an XOFF to the QTERM, and the user has pressed any keys, this command will clear the buffer, so that when the host sends XON to the QTERM, there will be nothing in the buffer for the QTERM to transmit to the host.

2.2.48 XON/XOFF Mode - ESC l

This command enables or disables the XON/XOFF operation of the QTERM.

Valid values for # are

@ - disable XON/XOFF operation
A - enable XON/XOFF operation

If you disable XON/XOFF operation, then any keys pressed by the user will be sent to the host immediately. If the host sends data fast enough to the QTERM to fill up the receive buffer, additional characters will be ignored.

2.2.49 User Area Read/Write - ESC m

This command allows you to store up to 63 bytes of your own information (such as serial numbers or parameters) in the current page of the user area of the QTERM nonvolatile EEPROM, then later read them from the terminal.

There are two valid values for #:

@ - read user data
A - write user data (followed by data)

READ DATA: if # = "@", the QTERM will transmit the data in the current page of the user area to the host in the following format:

. . . .

where # is a character in the range of 40h to 7Fh, and indicates the number of data bytes to follow (for multidrop units, # is in the range of 41h to 47h), and ". . . ." is the corresponding number of user bytes. These bytes will be exactly what was originally stored, so they may be any 8-bit value. If # = "@" (0 bytes to follow), then there was no data stored in the user area, i.e. the page is empty.

WRITE DATA: to write user data, use the format:

ESC m A # . . . .

where # is in the range of 41h to 7fh ("A" to DEL), and indicates that from 1 to 63 bytes of data are to follow (for multi-drop units, # is in the range of 41h to 47h). After receiving the number of bytes, the QTERM will return an ACK (06h) or a NAK (15h) to the host to indicate whether the string will fit in the current page. After the data is written to EEPROM, the QTERM sends another ACK or NAK to the host to indicate successful completion of the command. For multidrop units, the ACK or NAK is sent as a packet as soon as the write operation is complete.

Refer to Table 1-2 for a complete list of values of # and the corresponding number of data bytes for both the read and write commands.

2.2.50 Get Free User Area - ESC n

This command tells the QTERM to transmit one character to the host indicating the total size of the current page of the user area. This will always be 63 bytes (DEL character), unless you are accessing the very last available page, and it is a partial page.

Note that the value returned by this command does not tell you how much *more* you can write to the current page, but tells you how much *total* you can write to the current page.

This command is primarily for compatibility with the QTERM-II (not discussed in this manual); it will have limited usefulness for the QTERM-IV and the QTERM-P40, because there are always many complete pages available.

2.2.51 User Area Page Control - ESC o

This command allows you to access multiple data pages. You can get or set the current User Area page or you can obtain the total number of User Area pages available. See section 2.1.3 for more information about memory organization.

There are three valid values for #:

@ - Get the number of User Area pages

A - Set the current User Area page
B - Get the current User Area page

GET NUMBER OF USER AREA PAGES: This command tells the QTERM to return two ASCII characters (MSB and LSB) indicating the total number of User Area pages available.

SET CURRENT USER AREA PAGE: This command sets the currently active User Area page to the value specified by ## (MSB and LSB). *GET CURRENT USER PAGE:* This command tells the QTERM to return two ASCII characters (MSB and LSB) indicating the currently active page number.

2.2.52 Macro String Execution - ESC p

This command allows you to "execute" any of the 48 macro strings defined in the QDATA file (see Chapter 3 for more information on using the QDATA file). Since macro strings can contain commands to the QTERM, data for the host and/or data for the QTERM, a single macro string can configure almost any aspect of the QTERM operation. The value of # can vary from "@" (macro 00) to "o" (macro 47). For example, the string

ESC p A

will execute the macro string stored in the QDATA file under the "A" location.

2.2.53 Query Multidrop Buffer - ESC q

This command (ignored by non-multidrop units) causes the QTERM to return its multidrop buffer as the response packet. Do not send this command in the same packet with other response-generating commands, or as a part of a broadcast event.

If the multidrop buffer is empty, or this command is sent to the QTERM-IV during EDIT mode, the QTERM-IV will return a Null packet.

Table 1-2: Command Values for ASCII Characters.

Note: the column labeled "Time" refers to several commands:

Set Buzzer Duration	-T=duration of beep
Set Key Click Duration	-T=duration of key click
Set Key Repeat Rate	-2*T=period; 1 / (2*T) = rate
Set LED Blink Rate	-2*T=period; 1 / (2*T) = rate

ASCII Char	Hex Value	Decimal Value	Set Timeout Delay	Set Tab Spacing (spaces)	Query & Set Cursor Position (row col)	LEDs On/Off/Blink (led state)	Query Status (ds buf)	Time "T" (seconds)	Bar Code User Area String Length
@	40h	64	disable				on 0		1
A	41h	65	0:20				on 1	0.05	2
B	42h	66	0:40				on 2	0.10	
C	43h	67	1:00			shft off	on 3	0.15	3
D	44h	68	1:20			1 off	on 4	0.20	4
E	45h	69	1:40			2 off	on 5	0.25	5
F	46h	70	2:00			3 off	on 6	0.30	6
G	47h	71	2:20			4 off	on 7	0.35	7
H	48h	72	2:40	disable	0	5 off	on 8	0.40	8
				1	1				9
I	49h	73	3:00	2	2		on 9	0.45	10
J	4Ah	74	3:20				on 10	0.50	11
K	4Bh	75	3:40	3	3	shft on	on 11	0.55	
				4	4				12
L	4Ch	76	4:00	5	5	1 on	on 12	0.60	13
M	4Dh	77	4:20			2 on	on 13	0.65	14
N	4Eh	78	4:40	6	6	3 on	on 14	0.70	
				7	7				15
O	4Fh	79	5:00	8	8	4 on	on 15	.075	16
P	50h	80	5:20		0	5 on		0.80	17
Q	51h	81	5:40	9	1			0.85	
				10	2	9			18
R	52h	82	6:00	11		10		0.90	19
S	53h	83	6:20		3	11 shft blink		0.95	20
T	54h	84	6:40	12		1 blink		1.00	
				13		12			21
U	55h	85	7:00	14		13 2 blink		1.05	22
V	56h	86	7:20			14 3 blink		1.10	23
W	57h	87	7:40	15		4 blink		1.15	
				16		15			24
X	58h	88	8:00	17		16 5 blink		1.20	25
Y	59h	89	8:20			17		1.25	26
Z	5Ah	90	8:40	18				1.30	
				19		18			27
[	5Bh	91	9:00			19 shft toggle		1.35	28
\	5Ch	92	9:20			1 toggle		1.40	29
]	5Dh	93	9:40			2 toggle		1.45	
^	5Eh	94	10:00			3 toggle		1.50	30
_	5Fh	95	10:20			4 toggle		1.55	31
`	60h	96	10:40			5 toggle	off 0	1.60	32
a	61h	97	11:00				off 1	1.65	33
b	62h	98	11:20				off 2	1.70	34
c	63h	99	11:40				off 3	1.75	35

continued

Table 1-2: Command Values for ASCII Characters.

ASCII Char	Hex Value	Decimal Value	Set Timeout Delay	Set Tab Spacing (spaces)	Query & Set Cursor Position (row col)	LEDs On/Off/Blink (led state)	Query Status (ds buf)	Time "T" (seconds)	Bar Code User Area String Length
d	64h	100	12:00					1.80	36
e	65h	101	12:20					1.85	37
f	66h	102	12:40					1.90	38
g	67h	103	13:00					1.95	39
h	68h	104	13:20					2.00	40
i	69h	105	13:40					2.05	41
j	6Ah	106	14:00					2.10	42
k	6Bh	107	14:20					2.15	43
l	6Ch	108	14:40					2.20	44
							off 4		
m	6Dh	109	15:00				off 5	2.25	45
n	6Eh	110	15:20				off 6	2.30	46
o	6Fh	111	15:40					2.35	47
							off 7		
p	70h	112	16:00				off 8	2.40	48
q	71h	113	16:20				off 9	2.45	49
r	72h	114	16:40					2.50	50
							off 10		
s	73h	115	17:00				off 11	2.55	51
t	74h	116	17:20				off 12	2.60	52
u	75h	117	17:40					2.65	53
							off 13		
v	76h	118	18:00				off 14	2.70	54
w	77h	119	18:20				off 15	2.75	55
x	78h	120	18:40					2.80	56
y	79h	121	19:00					2.85	57
z	7Ah	122	19:20					2.90	58
{	7Bh	123	19:40					2.95	59
|	7Ch	124	20:00					3.00	60
}	7Dh	125	20:20					3.05	61
~	7Eh	126	20:40					3.10	62
del	7Fh	127	21:00					3.15	63

CHAPTER 3.

USING THE QDATA FILE

3.1 QDATA Files

A **QDATA** file is used to configure your terminal. There are four reasons you may want to use a QDATA file:

- You want to change what keys send when they are pressed, or design your own display characters (this *must* be done with the QDATA file).
- You want to change some of the other defaults (such as beep length, tab settings, etc.).
- You want to define event-driven strings or macro strings.
- You want to protect your development investment by modifying the terminal from its factory-default operation.

If the fourth reason is important to you, be sure to consider modifying the <*qdata retrieve mode*> identifier to ensure that your investment remains protected.

With your terminal you will have received a disk (PC-compatible) with a variety of files. Two of these files are

QDATA File	An ASCII (text) file that specifies values for all of the configurable parameters of the QTERM.
QSETUP.EXE	A program which loads the QDATA file into the QTERM.

There are actually several versions of the QDATA file on the disk:

QDATA24.Vxy	For 24-key terminals.
QDATA40.Vxy	For 40-key terminals.
QDATA24W.Vxy	For 24-key terminals with wide-temperature displays.
QDATA40W.Vxy	For 40-key terminals with wide-temperature displays.

The file extension “Vxy” will actually be the QDATA software version, represented by “Version x.y”. For example “QDATA24.V21” would be the QDATA file for a 24 key QTERM with version 2.1 QCODE.

You will only need to use the QDATA file that matches the hardware on your terminal. The QDATA file for your QTERM is already set up with default values for all parameters. You will modify only the parameters which you want to change from their default. NOTE: The QTERM-P40 is not available with a wide-temp display.

Your QTERM is delivered with the default QDATA file already installed. You only need to modify the appropriate QDATA file and load it into the terminal if you want to change any of the default values. If not, you can skip this entire chapter.

Note that all QTERM parameters can be modified using software commands (see Chapter 2) *except for key strings and custom display characters!* The only way to modify what a key sends when it is pressed, or to redefine the custom display characters, is by modifying the QDATA file and loading it into the QTERM using the QSETUP program.

3.2 Using QSETUP

Your distribution disk includes the QSETUP.EXE program, which is used to load the QDATA file into your QTERM. This program actually performs three tasks:

- load a QDATA file into the QTERM
- retrieve a QDATA file *from* the QTERM
- load the operation code file (QCODE file) into the QTERM

To use QSETUP, you *must* use power-on setup to set the QTERM's baud rate and data format to 9600/8n1. These are the only settings that will work with QSETUP.

You must also connect your terminal to your PC's COM port and supply power to it. If you do not have an EIA-232

terminal, you will need an interface unit. See Chapter 1 for more information.

When you first run QSETUP, you will see an introductory screen that shows the version number. This version *must* match the version of the QDATA file you are going to load. See the *Versions* section below for more information.

The next screen will give you a choice of COM ports, if your PC has more than one. Select the port to which you have connected your terminal.

3.2.1 Loading a QDATA File

To load a QDATA file from disk into the QTERM, select the *Load QDATA* menu item. You will be asked for the file name. Enter the file name, along with the extension, then press the *Enter* key.

QSETUP will now open the specified file, scan it for errors, and, if none are found, load it into the QTERM. This will take about 10 seconds.

If any errors are found, QSETUP will display an error message indicating the problem. Use your editor to fix the problem, then try to load it again.

3.2.2 Loading a QCODE File

Also included on your distribution disk is a file called QCODE.V21. This is the actually microprocessor code for running the QTERM. If the code in the QTERM somehow gets corrupted, you can load this file into the terminal to get it running again.

If you suspect your terminal has corrupted code, call Beijer Electronics for technical assistance in reloading the QCODE file.

3.2.3 Retrieving a QDATA File

If you select the third menu item on the QSETUP menu screen, you will be asked for a file name into which QSETUP will put the retrieved QDATA information. Specify a file name that does not exist.

QSETUP will retrieve the QDATA information from your terminal and put it into a disk file. This file is similar to the default files supplied with your terminal, except for two changes:

- The retrieved file will have no comments in it.
- The retrieved file will have exactly one identifier per line, so it will have many more lines than the default QDATA file (which puts several identifiers on one line in the key sections).

Since neither of these affects the parameter content of the retrieved QDATA file, this retrieved file can be downloaded into any QTERM without modification.

3.3 Basic QDATA File Concepts

Identifiers. The QDATA file contains "identifiers" used by QSETUP to determine how to configure the QTERM. These identifiers appear in two formats:

[....] and <....>

The most important rule about editing the QDATA file is: NEVER DELETE OR EDIT ANY OF THE IDENTIFIERS, AND NEVER CHANGE THEIR ORDER!

You will edit the *values* assigned to the identifiers, but you may not edit the identifiers, delete the identifiers or change the order in which they appear.

Versions. You *must* use the same version of the QDATA file and the QSETUP program as the version of code in your QTERM.

- Your QDATA version is noted in the file name extension (i.e. QDATA40.V21 is version 2.1) and is the first identifier in the QDATA file.
- The QSETUP version is displayed on the first screen when you start the QSETUP program.
- The version of software in your QTERM is displayed if you hold down any key and apply power.

If you find that you have different QDATA, QSETUP and QTERM code versions, contact Beijer Electronics for updated software and instructions. Do *not* attempt to use different versions of these items together; you will probably corrupt the QTERM operational code.

QDATA File Size. The QDATA file which you edit can be as large as you want. When QSETUP sends the QDATA information to the QTERM, it only sends actual parameters and key strings. The maximum amount of information from the QDATA file which can be stored by the QTERM is about 20 kbytes, which should be enough for even the most memory demanding applications.

The default QDATA files generate about 300 bytes of information, so you can add over 19 kbytes of data to key strings before you exceed the 20 kbyte limit. QSETUP will tell you if there is too much data to be stored by the QTERM.

ASCII Editor. You may use any editor to modify the QDATA configuration file, but the output from the editor must be in ASCII format, without any embedded codes.

Included on your QTERM distribution disk is the TED3 editor (TED3.COM), which you can use to edit the QDATA file. The file TED3.DOC provided documentation for the editor and its commands.

Uppercase/Lowercase. For quoted text strings, (used for key strings), the string is assigned exactly as shown in the QDATA file.

Comments. Comments may be added to the file as desired. A comment starts with a semicolon, and ends at the end of a line (at the carriage return).

Whitespace. Whitespace may be added to the file as desired. This means you can add spaces, tabs, carriage returns and line feeds anywhere you would like. The only exceptions are identifiers (such as [shift mode]) with whitespace and in quoted text strings (which are assigned to keys); anything within the quotes will be assigned to the key.

Since you can add and remove carriage returns, you can place each identifier on a separate line, or you can have several identifiers on one line, as long as the order in which they appear is not changed.

3.4 Parameter and Key String Sections

The QDATA file has two sections: the *parameter section* at the top, followed by the *key string section* at the bottom.

3.4.1 Parameter Section

In the parameter section, you will assign a number to each identifier.

Numbers. The number can be specified in one of two ways:

- decimal numbers (49, 27)
- hexadecimal numbers (x0D, x8C)

These numbers are always positive integer numbers within the range specified by the comments. Hexadecimal constants start with the letter “x”. Decimal and hexadecimal constants are equivalent; use whichever one is easier at any time.

3.4.2 Key String Section

This section follows the parameter section. The format for the identifiers for each key is

<**k**#*> - unshifted key strings
<**sk**#*> - shifted key strings
<**kr**#*> - unshifted key release strings
<**skr**#*> - shifted key release strings

where:

= row (top = row 0)
* = column (right = column 0)

Examples of this notation:

<k21> is for unshifted key string for row 2 and column 1

<skr32> is for shifted key release string for row 3 and column 2.

Since the QTERM electronics can support up to eight rows and six columns of keys, there are identifiers for 48 unshifted and 48 shifted key strings, as well as identifiers for 48 unshifted and 48 shifted key release strings. All of these must remain in the file; however, you will have no reason to assign key strings to keys that do not exist on your keypad.

In the key string section the type of values you can assign to a key are

- strings enclosed in single quotes ('abcde')
- decimal numbers (49, 27)
- hexadecimal numbers (x0D, x8C)
- Special Function Switches (\L, \P, \S)
- a combination of the above, separated by commas

Strings. Text strings can contain any printable ASCII characters and are enclosed in single quotes. If you need to include a single quote in your text string, terminate the string and specify the ASCII value as a number (x27 hexadecimal or 39 decimal).

Numbers. Numbers are always positive integer numbers within the range 0 to 255. Hexadecimal numbers start with the letter “x”. Decimal and hexadecimal numbers are equivalent; use whichever one is easier at any time.

Special Function Switches. These special switches are used for advanced functions such as Local Mode toggle,

pausing and Edit Mode control. See section 3.5.3 and section 4.3 for a detailed description of these switches.

Examples of Key Strings. The simplest example of a key string is where pressing the key sends just one character, regardless of whether the shift key has been pressed:

```
<k43> '1'  ; send ASCII '1'
<sk43> '1'; send same if shifted
<kr43>     ; send nothing on release
<skr43>    ; send nothing on shifted release
```

The next example shows a key which sends a string when pressed, but beeps if pressed when shifted (the number 7 is a “bell” character):

```
<k27> 'Hello There'; send string
<sk27> 7  ; beep if shifted
<kr27>     ; nothing
<skr27>    ; nothing
```

Below is a key string which uses Special Function Switches to control the keypad LEDs and to pause. The key controls LED1 when unshifted (press and the LED goes on, release and it goes off), and controls LED2 when shifted (press and it goes on, release and it goes off ½-second later):

```
<k32>     'Turn 1 on',\L,27,'PI';send string, turn L1 on
<sk32>    'Turn 2 on',\L,27,'PJ'; send string, turn L2 on
<kr32>    'Turn 1 off',\L,27,'PA' ; send string, turn L1
          off
<skr32>   'Turn 2 off',\P,10,\L,27,'PB'; send string,
          pause ½-second, turn L2 off
```

The final example shows how to combine strings, numbers and Special Function Switches on one key string assignment. Note the splitting the key assignment onto multiple lines (as done here) is perfectly acceptable.

```
<k62> 'Pause here ',\P,10,'then turn dig0 on ',
      \L,27,'dH','then beep ',7,'then send a cr/lf '
      'to the host',x0D,10
```

Key String Restrictions. A key string may be from 0 to 255 bytes long.

Pause constants may be from 1 to 255, which corresponds to 50 ms to 12.75 second pauses. Multiple pause constants can be specified in one string.

Be sure to separate each component of a key string (text strings and numbers) by a comma, as shown in the examples above.

3.4.3 Macros and Event Triggered Strings

This is the last section of the QDATA file, and contains macro strings and strings that get executed according to certain events. The following section explains these advanced features in more detail.

3.5 Advanced QDATA File Concepts

The QTERM has some advanced capabilities in its string processing ability that go beyond simple byte strings that get transmitted to the host.

3.5.1 String Terminology

The various string definitions available in the QTERM provide a great deal of flexibility when interfacing to a variety of host systems. In order to understand the operation of QTERM strings, you must understand the following terminology:

STRING DEFINITION
A string of bytes specifying the data to be transmitted or displayed and any commands to be executed when the defined string is executed. *String Definitions* can only be entered in a QDATA file. The program QSETUP is then used to compile the QDATA file and download it into the QTERM's EEPROM memory.

STRING EXECUTION
The transmission or display of characters, or execution of software commands specified in a given *String Definition*. Strings can be executed in a number of ways. For example, by pushing a key (this would execute the string defined for that key) or by undergoing a timeout (this would execute the timeout string).

HOST MODE
Forces *String Definition* data to be transmitted to the host. All *String Definitions* are executed in *Host Mode*, by default, unless the string execution mode is switched by a *LocalMode toggle*. *Host Mode* only applies to *String Definitions*; it has no meaning outside of a *String Definition*.

LOCAL MODE
Forces *String Definition* data to display or execute locally on the QTERM. Software commands and displayable data are handled as though they were received from the host. *Local Mode* only applies to String Definitions; it has no meaning outside of a *String Definition*.

Local mode is enabled by entering the *Local Mode toggle* (“\L”) in a *String Definition*. Any data following the *Local Mode toggle* is interpreted as *Local Data* by the QTERM

up until another *Local Mode toggle* is encountered, after which *Host Mode* becomes active again. *Local Mode* can be toggled on and off as much as desired within a *String Definition*.

HOST DATA
That portion of a *String Definition* which is executed in *Host mode* and specifies what is to be transmitted to the host. Since *Host Mode* is the default String Definition mode, *String Definitions* are always executed as *Host Data* until a *Local Mode toggle* is encountered.

LOCAL DATA
That portion of a *String Definition* which is executed in *Local Mode* and specifies what is to be displayed or executed on the QTERM-IV itself when a given string executes. *Local Data* may contain *Display Data* or *Local Commands*.

DISPLAY DATA
That portion of either an incoming packet or a *Local-Mode* string definition which contains data for display on the LCD.

LOCAL COMMANDS
That portion of a *String Definition*, executing in *Local Mode*, which contains QTERM software commands which are to be executed on the QTERM. Local Commands can only consist of QTERM software commands. Certain commands are not allowed as Local Commands; certain other commands take on special behavior as Local Data when the QTERM's serial interface mode is set to Block or Multidrop mode, i.e. EDIT buffer editing keys. This special behavior is described in the Block and Multidrop Serial Mode sections below.

3.5.2 String Concepts

The QTERM takes the concept of a "character string" and broadens it to include any combination of these items:

- Host data (text to be transmitted to the host)
- Local display data (text to be displayed on the QTERM screen)
- Local commands (commands to be executed as if sent from the host)
- Special functions including Pauses (delays) in string processing and Multidrop/Block mode buffer control

A string is "executed" when it is processed by the QTERM. The concepts of a "string" and of "string execution" apply to all references to strings on the QTERM:

- key strings (executed when a key is pressed)
- key release strings (executed when a key is released)
- macro strings (executed when a the macro command is received)
- autoexec string (executed on power-up or reset)
- timeout string (executed when a timeout occurs)
- low-battery string (executed when the lowbat line goes low)

3.5.3 String Definitions

String definitions describe what happens when a given string is executed. String definitions can only appear in QDATA files. Then you use the QSETUP program to compile the QDATA file and all the string definitions and other configuration data contained in the QDATA file and download it into the QTERM. String Definitions are constructed using a combination of quoted text (**'text'**), numerical byte values (**27**, **xF3**) and three optional special-function switches: **\L**, **\P** and **\S**.

The Local Mode Toggle switch, **\L**, toggles the string between host and local mode. For example, here is a QDATA file key-string definition which uses Host Mode to transmit a string to the host and Local Mode to display a different string on the QTERM's LCD.

<k13> 'Host',\L,'QTERM'

This would send the text "**Host**" to the host computer *(host data)* and the text "**QTERM**" to the QTERM display *(local-mode display data)* when the key at row #1 and column #3 is pressed. The **\L** can be used multiple times in a string to toggle back and forth between host and local modes. All strings always default to host data until a **\L** is encountered.

One of the useful features of local mode is to cause the QTERM to execute software commands without *receiving* the commands from the host. For example, if you defined the autoexec string as

[autoexec string] '!',\L,'QTERM Active', 07,x1B,'IC@'

on power up or reset, the QTERM would send "**!**" to the host and display "**QTERM Active**" on the LCD. Then the QTERM buzzer will beep, since **07** is the ASCII BEL code, and the cursor would move the first column on the last row since **ESC IC@** is the command sequence to position the cursor at row 3 and column 0.

The Pause switch, **\P**, inserts a pause in string execution. The form is **\P,n** where **n** is a number indicating the number of 50 ms periods to pause. For example:

<k00> 'First',\P,20,'Second'

would send the text "**First**" to the host, pause for one second (20*50 ms = 1 sec), then send the text "**Second**" to the host. **n** can be a decimal number (1 to 255) or a hexadecimal number (x01 to xFF).

The EDIT Buffer Start switch, **\S**, sets the start of the Multidrop/Block EDIT buffer at the current cursor location *(this is only useful when the QTERM is in either Multidrop or Block mode)*. The EDIT buffer extends to the end of the current line. A more detailed description of Multidrop and Block modes and the EDIT buffer follows below.

Any of the special function switches can appear anywhere inside a string definition.

3.5.4 String Examples

Example 1: A More Sophisticated Warning Key String

<k00> 07,'ALARM!',\L,27,'E','WARNING!!',
27,'PQ',07,\P,20,07

Pressing this key would result in the following:

07	-sent to host, should cause host to "beep"
'ALARM!'	-sent to host for display
\L	-switch to local mode
27,'E'	-clear screen command to QTERM (27 = ⇜)
'WARNING!!'	-displayed on QTERM screen
27,'PQ'	-command to blink rightmost QTERM LED
07	-tells QTERM to beep
\P,20	-pauses for 1 second
07	-tells QTERM to beep again

Example 2: A Macro String for Displaying a Menu *(make sure AutoLinefeed is ON for this one!)*

<m00>L,27,'E',
' Please Select One:',x0D,
' 1) Main Menu ',x0D,
' 2) Setup Menu ',x0D,
' 3) Text Menu '

If the host sent the command **⇜ p @** (to execute macro 00) to the QTERM, the QTERM would clear its screen (⇜ E), then display the menu text shown above. All of this with one three-byte command from the host!

Example 3: Key Strings Executing Macro Strings

Presume that the macro above is defined, and that we define a key as follows:

<k00> 'Entering Menu',\L,27,'p@'

If the user presses this key (which we might want labeled "MENU" on the graphic overlay), the QTERM will send the string "**Entering Menu**" to the host, and then execute Macro 00, which displays the menu shown above. All without any assistance from the host!

3.6 Custom Display Characters

The QTERM is capable of displaying eight user-defined characters. Each of these characters consists of an eight row by five column dot matrix, where you specify which of the dots are on or off.

You also define what code will cause the character to be displayed by the QTERM. You may choose any 8-bit value (code) from 00h through FFh, *except* values from 07h through 1Bh (see Appendix B for a chart of this). This allows you to redefine up to eight of the standard ASCII character bit patterns. For example, you could redefine the hex value x62, which normally displays a "b", to display a small box (□).

Each custom character is defined in the QDATA file as a string of eight byte values, where each byte specifies the dot pattern for one row of dots. For each byte, the upper three bits are set to 0, and the lower five bits correspond to the five dots on the row. If a bit is set to 1, the dot will be dark (on); if the bit is set to 0, the dot will be light (off).

As an example, Figure 3-1 shows the default dot pattern of custom character 6 which is a little humanoid figure. From this dot pattern, the following bit assignments are derived:

Row	7	6	5	4	3	2	1	0	Hex
1	0	0	0	0	1	1	1	0	0E
2	0	0	0	0	1	1	1	0	0E
3	0	0	0	0	0	1	0	0	04
4	0	0	0	1	1	1	1	1	1F
5	0	0	0	1	0	1	0	1	15
6	0	0	0	0	0	1	0	0	04
7	0	0	0	1	1	1	1	1	1F
8	0	0	0	1	0	0	0	1	11

As you can see in your default QDATA file, the hexadecimal values listed in the right column above are the values which are used to define the custom character. This particular dot patter is assigned to the ASCII character x06 in the default QDATA file.

Default dot patterns and character assignments for all eight custom characters are shown in the chart in Appendix B.

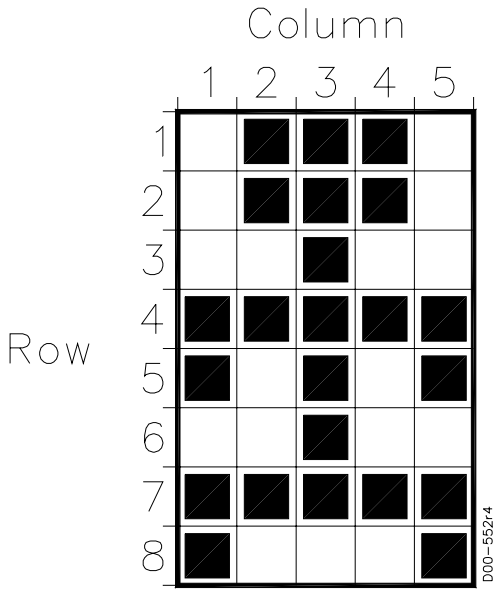


Figure 3-1. Example Custom Character.

CHAPTER 4.

QTERM SERIAL MODES

4.1 Serial Modes

Regardless of the serial interface *hardware* selected, the QTERM *software* can operate in three different serial interface modes: Full-Duplex, Block and Multidrop Modes. These serial modes can be selected from within either Power-On Setup mode or the QDATA file by setting the **[serial mode]** field.

Not all serial modes are useful with all of the different types of serial interface hardware available with the QTERM. For example, Full-Duplex mode is not useful with an EIA-485 hardware interface.

Block Mode is very similar to Multidrop Mode. If your QTERM is a Multidrop unit, we recommend that you read the Block Mode section before reading the Multidrop section.

4.2 Full-Duplex Serial Mode

When in Full-Duplex serial mode, the QTERM immediately executes key strokes as they are pushed. Host mode string data is sent to the host and local mode data, and commands and special functions are executed when the key is pressed. Commands are executed and displayable data is displayed as soon as they are received from the host. Only one terminal may be connected to the host.

Full-Duplex may be used only with the following hardware interfaces: **EIA-232, -422, and 5-volt Buffered**.

Xon/Xoff handshaking is supported and is *only* available in Full-Duplex mode. If desired, Xon/Xoff handshaking can be disabled from within the QDATA file.

4.3 Block Serial Mode

Block Serial Mode is useful for prompt/response systems with semi-intelligent hosts where the user must construct a complete, correct response at the QTERM, prior to transmission, without host intervention.

In Block Mode, all communication between the Host and the QTERM is done with data packets. A valid Block Mode data packet consists of a data portion followed by a user-definable, End-Of-Packet (EOP) terminator character. The data portion may contain any number of displayable or command data bytes, but must not contain a byte equal to the EOP character. The EOP character is user-definable in the QDATA file; any 8-bit value is valid.

Block Mode may be used only with the following hardware interfaces: **EIA-232, -422, and 5-volt Buffered**.

Since Block Mode is not a full-duplex scheme, the QTERM cannot simultaneously process incoming data from the host *and* keystrokes. This does not mean that data will be lost, but it does mean that the host software must support the communications scheme described below.

In order to understand the operation of Block Mode *(and Multidrop Mode)*, you must understand the following terminology:

DATA PACKET
A formatted sequence of bytes which *must* be used to transmit data to and from the QTERM running in Block Mode. The format is very simple. All *Data Packets* consist of a free-form data portion followed by an *EOP Terminator* (described below). The data portion consists of a stream of bytes and may be as long as desired, but it may not contain a character equal to the *EOP Terminator*. The data portion must always be followed by the *EOP Terminator*.

EOP TERMINATOR
A special character, defined in the *QDATA file*, which performs two functions. It terminates all *Data Packets* sent from and received by the QTERM when in Block Mode. It is also used to terminate *EDIT Mode* and causes the *EDIT Buffer* to be sent to the host when a key, defined with the *EOP Terminator* character, is pushed. If you wish to use Block Mode, you must define at least one key with the *EOP Terminator*.

The QTERM automatically sends the *EOP Terminator* at the end of all *Response Packets* it returns to the host.

RESPONSE PACKET
A *Data Packet* automatically returned by the QTERM after it receives a complete transmission from the host. The *Response Packet's* data portion can consist of nothing (a NULL packet), data returned in response to a query command or an acknowledgment byte. The *Response Packet* is always terminated with an *EOP Terminator*.

IDLE MODE
The operating mode of the QTERM anytime it is neither processing strings nor receiving data from the host.

EDIT MODE
The operating mode in which string data is entered into an LCD buffer and edited prior to transmission to host.

4.3.1 Receiving Data from the Host

If the QTERM is in IDLE, and the host begins a transmission, the QTERM will *only process data coming from the host* until the EOP Terminator is received; keystrokes will be buffered but not processed. While receiving the incoming packet from the host, the incoming data is displayed and incoming commands are executed as they are received.

Upon receiving the EOP Terminator from the host, the QTERM will send a response packet, process any buffered keystrokes, and revert to IDLE to await the next event.

4.3.2 Response Packets

The QTERM *always* returns a response packet after receiving a host packet (unless ACK/NAK responses are disabled; see below). The data portion of the response packet will always consist of one of the following: nothing (a Null Packet), query data (in answer to a query command sent by the host), or either an ACK (ASCII 06h) or NAK (ASCII 15h) character. Response packets are always automatically terminated by the EOP character.

Null Packets are only returned in response to a Query Buffer command; this command is not useful in Block Mode since the EDIT Buffer never needs to be queried. Other types of query commands sent by the host will cause the QTERM to return data.

If the QTERM receives a command from the host which requires that the QTERM return data to the host, i.e. a Query Cursor Position command etc., the requested data will be sent in lieu of any acknowledgment byte.

Acknowledgment bytes are only returned if nothing in the incoming host packet requires the QTERM to return query data. For example, if the host sends a packet containing only display data to the QTERM, the data portion of the QTERM response packet will contain only an ASCII ACK or NAK acknowledgement byte. If instead, the host sends display data and a Query Cursor Position command, the QTERM response packet will contain the current cursor position only without an acknowledgement byte.

The only acknowledgement bytes are ACK (ASCII 06h) or NAK (ASCII 15h) character to indicate that the QTERM received the host's transmission without error or that the QTERM was busy and ignored the host transmission, respectively. Automatic transmission of response packets containing acknowledgement bytes may be disabled, if necessary, by the **[MD/Block response mode]** field in the QDATA file.

4.3.3 Transmitting Data to the Host and the Operation of EDIT Mode

The host data portion of a string is not sent immediately to the host when the string is executed. Instead, host data is buffered on the display in a special area called the EDIT buffer. The EDIT buffer position is variable and is reset each time the QTERM receives a packet from the host. At the termination of an incoming packet, the start of the EDIT buffer is relocated to the last cursor position. The EDIT buffer always extends to the end of the same line on which it started.

If a key is pressed while the QTERM is in IDLE, or if there are buffered keystrokes waiting to be handled after the QTERM received a packet, or some other string has been triggered (i.e. Timeout string, Autoexec string etc.) the QTERM enters EDIT mode.

While in EDIT mode, the *host data* portion of executed strings will be entered into the EDIT buffer up to the end of the buffer. Keys may be defined using certain commands as edit keys (described below) with which to modify the data in the EDIT buffer prior to transmission. While in EDIT mode, incoming packets from the host will not be processed *(with a few exceptions noted below)* but will generate a response packet containing a NAK acknowledgement indicating to the host that the packet was not processed.

Keystrings will continue to be appended to the EDIT buffer until the EOP Terminator is encountered in the current string definition or until the user hits a key defined with the EOP character. When the EOP character is encountered, the entire EDIT buffer is sent to the host, the cursor is returned to the start of the EDIT buffer, EDIT mode is terminated and the QTERM returns to IDLE. Because of this, you must

have at least one key defined with the EOP character. The EDIT buffer is not cleared automatically after transmission.

4.3.4 EDIT Mode and the Use of Local and Host Data

Because host data in the string definitions is sent to the EDIT buffer on the LCD, Local-mode, display data loses some of it's utility when you run in Block Mode. For example, a key defined as follows:

<k00> 'To Host',\L, 'Local data'

would put both "**To Host**" and "**Local Data**" in the EDIT buffer. Since both strings wind up in the EDIT buffer, what use is Local Mode? Don't be fooled; there are big differences in the way Host and Local data are handled.

Host data is not interpreted in any way by the QTERM, other that to detect an EOP Terminator indicating the EDIT buffer should be transmitted. The QTERM simply puts all Host Data, *as is*, into the EDIT buffer. On the other hand, Local Data is sent through the QTERM's command processor. The command processor looks for any QTERM commands and executes them. If it cannot find any commands, it assumes the Local Data is just display information and sends it to the LCD. Both the strings in the key definition above get into the EDIT buffer, but they get there in entirely different ways.

Because of this behavior, you probably will not use Local Mode *display data*, by itself, very often while running in Block Mode. You *will* find uses for Local Mode *commands* though. For example, by executing a Local Mode cursor positioning command, you could put Local mode display data on the LCD but somewhere other than the EDIT buffer. You cannot do this with Host Data; it is *always* directed to the EDIT buffer.

4.3.5 Defining Editing Keys for the EDIT Buffer

Certain commands, when executed from within a Local Mode string while running in Block Mode, can be used to create EDIT Buffer editing keys so the user can modify characters entered into the EDIT buffer prior to sending the buffer to the host. These commands, if executed from within a Local Mode string, behave differently than if they were received from the host and operate *only on the EDIT buffer*. These commands are: BackSpace (ASCII BS, 08), DELete (ASCII DEL, 127), Cursor Left (ESC D), Home Cursor (ESC H), and Flush TX buffer (ESC k).

Backspace, Delete and Cursor Left will only move the cursor left to the start of the EDIT buffer. Home Cursor homes the cursor to the start of the EDIT buffer and Flush TX Buffer clears the EDIT buffer and returns the QTERM to IDLE mode. These functions are not usually limited to operating on the EDIT buffer; this behavior only occurs when they are executed within a Local Mode string while the QTERM is in Block Mode.

As an example, you might want to define 3 EDIT buffer editing keys: clear buffer, cursor left and cursor right. These keys would be defined as follows:

```
<k00> \L,x1B,'H',x1B,'K'; Clear Buffer
<k01> \L,x1B,'C'; Cursor Right
<k02> \L,x1B,'D'; Cursor Left
```

There are a few things to notice in these definitions. First, note that the Clear Buffer key *does not* use the **Flush TX buffer** command but rather, homes the cursor (ESC H) then clears to the end-of-line (ESC K). This is because the **Flush TX Buffer** command not only clears the EDIT buffer, it also returns us to IDLE mode. To clear the EDIT buffer and remain in EDIT mode, home the cursor to the start of the buffer with the ESC H command, then clear the line.

Also note that the Cursor Right key uses the standard Cursor Right command which is not one of the commands mentioned above which takes on special behavior regarding the EDIT buffer. This is because the Cursor Right command naturally stops at the end of the current line which is also always the end of the EDIT buffer.

Lastly, the Cursor Left key uses the standard Cursor Left command. This command will normally move the cursor left up to the beginning of the current line. In this application though, since we are in Block Mode and it is being executed from within a Local Mode string, the Cursor Left command takes on special behavior and stops at the start of the EDIT buffer rather than the start of the current line.

4.3.6 The EOP Terminator and EDIT Mode

In order to transmit the EDIT buffer to the host, the QTERM must execute a string defined with the EOP Terminator character. When the EOP Terminator is executed, the EDIT buffer is transmitted, the cursor returns to the start of the EDIT buffer, EDIT mode is canceled and the terminal returns to IDLE. Note that the EDIT buffer *is not cleared*.

If you wish to clear the EDIT buffer after it is transmitted, you can define your "Enter" key as follows:

```
<k53>x0D,\L,x1B,'K';Transmit buffer
          ;and clear it
```

x0D is the default EOP Terminator (you may define EOP to be any byte value), then we switch to Local Mode and clear

to the end of line with **x1B,'K'** (ASCII ↑ K). This works because anytime the QTERM executes the EOP and transmits the EDIT buffer, it will return the cursor to the start of the buffer automatically.

The EOP Terminator can be entered separately from the other data entered into the EDIT buffer, or it can be appended to a string definition. If EOP is appended to a string definition, the string will be entered into the EDIT buffer and be transmitted in a single step. For example,

[timeout string]
'timeout!',x0D,\L,x1B,'K'

would put "**timeout!**" in the EDIT buffer, transmit it (x0D = EOP), and then clear the EDIT buffer. All this would happen so fast, you would not see anything on the LCD.

4.3.7 Received Commands Allowed in EDIT Mode

The contents of most incoming packets are ignored by the QTERM if it is currently in EDIT mode; in these cases the QTERM will return a NAK acknowledge packet to the host in response. Certain commands, though, are received and processed by the QTERM, even when it's in EDIT mode. These commands are: Reset, Sleep, Query Status, Set Keypad mode, TX buffer flush and Query buffer.

These commands allow the host to take control of the QTERM regardless of its current operating mode.

4.3.8 The Special Function Switch \S and EDIT Mode

In some situations, you may want to reset the start of the EDIT buffer from within a string definition. The EDIT Buffer Start switch, **\S**, sets the start of the Multidrop/Block EDIT buffer at the current cursor location and may appear anywhere inside a string definition.

For example, you might want to set up a function key which displays a prompt and sends a prefix peculiar to the host to notify it of a parameter to be entered later by the user:

<k31> \L,x1B,'E','Enter Speed:',\S,\L,
- sp',x0D,\L,x1B,'H',x1B, 'K'

This one is complicated, but let's see how it works. When the key string executes, we first switch to Local mode to clear the screen and position the cursor to the top left. Then we display the prompt, "**Enter Speed:**" on the LCD and reset the EDIT buffer to the location just after the prompt with the **\S** switch.

Next, we toggle back to host mode with another **\L** switch and send to the EDIT buffer the "**-sp**" prefix, which the host will recognize as a command to accept a speed parameter. Then the default EOP Terminator, **x0D**, is encountered so "**-sp**" is sent to the host. Remember, the prefix is sent to the host in a packet which is terminated by the EOP Terminator. *(This "prefix" is only useful to the mythical host used in this example; it has no significance to the QTERM.)*

Lastly, we toggle back to Local Mode with another **\L** switch, home the cursor back to the start of the EDIT buffer (↑ H) and clear it out (↑ K) so the user can enter the speed parameter. *This happens so fast, the user will never realize that the prefix has been sent to the host.*

Without the **\S** switch, you could not separate the prompt from the prefix and both would be sent back to the host.

4.3.9 Auxiliary Serial Port Operation

Section 2.2.46 gives an explanation of the Auxiliary Port Control Command, and the general operation of the auxiliary port. There are a few things to note about operating the auxiliary port while in Block Mode.

Data packets, as described above, are never used for communications between the aux port and devices connected to it; this is true for Block Mode or for any other serial mode. The host can disable the port, initiate transmissions from the aux port, or it can enable reception from the aux port but it can only do one of these at a time; i.e. you cannot simultaneously transmit and receive on the aux port. The port disable command and port-transmission enable command operate as described in the User's Manual, but reception from the port is a bit different than that described in the User's Manual when the QTERM is in Block Mode.

To understand how received aux port data is handled in Block Mode, think of the aux port as a secondary keypad device. All data received at the aux port is handled by the QTERM just as though the data was entered on the keypad as would Host Mode key strings. This means that incoming data *is not* sent immediately to the host as it would be in Full-Duplex Serial mode. Instead, the QTERM enters EDIT mode and stores all incoming data in the EDIT buffer. Incoming aux port data is entered into the buffer until the EOP Terminator is received from the aux port or entered on the keypad, then the EDIT buffer is sent to the host.

Keystrokes and other executed strings may be freely mixed with aux port input in the EDIT buffer. Aux port data may be edited just like all other data stored in the EDIT buffer. The Aux Port Control Command can also be executed in Local Mode from within a string definition.

4.4 Multidrop Serial Mode

Multidrop Mode is a *half-duplex* communications mode which allows you to connect up to 32 QTERMs to one host port. Only one host is allowed in any network, and only the host can initiate network communications. Each terminal has a unique address which is used to direct data packets from the host to a single QTERM on the network. Please read the Block Mode sections above before continuing; Multidrop operation is *very* similar to Block Mode operation and the following discussion will explain Multidrop operation in terms of its differences from Block Mode operation.

Multidrop Mode may be used only with the following hardware interfaces: **EIA-422 and EIA-485**. This makes EIA-422 is the most flexible interface, in terms of serial modes since you can use any Serial Mode with the EIA-422 hardware interface.

All communications between the host and the QTERMs is done with Data Packets. Multidrop Data Packets are similar to those used in Block Mode with the addition of two address bytes at the start of the packet. The format of Multidrop data packets is described completely below.

In order to understand the operation of Multidrop Mode, you must understand the following terminology:

PENDING MODE
The operating mode in which string data, previously buffered in EDIT mode, is held until queried by the host. The QTERM can only enter *PENDING Mode* after passing through *EDIT Mode*.

The QTERM enters *PENDING Mode* when it encounters the *EOP Terminator* while processing string data. When this happens, instead of transmitting the EDIT buffer to the host, as would happen in Block Mode, the QTERM enters *PENDING Mode*, the keypad is disabled, and the LCD blinks to indicate to the user that no more data may be entered and that the QTERM is waiting for a query from the host.

The host must then query the QTERM for the contents of the EDIT buffer. When queried, the QTERM transmits a packet containing the buffer contents, the keypad becomes active, the LCD stops blinking, and the terminal reverts to IDLE mode.

TURN-AROUND TIME
Multidrop turn-around time is the time between the reception of the last byte of an incoming packet to when the terminal switches on its transmitter to send a response packet. This time is programmable on the QTERM via a setting in the QDATA file.

BROADCAST PACKET
A data packet sent by the host which is prefixed with the unit address “00” ; *all* terminals on the network will process this packet. Because all the terminals process *Broadcast Packets*, no terminal is allowed to return a response packet in answer to a *Broadcast*.

4.4.1 Multidrop Drivers and the Hardware Interface

The EIA-422 interface is not normally used as a multidrop network, but it can be if only one host is allowed per network. The host does not need tri-state transmit drivers since the host's transmitter is only connected to the QTERM's receiver and never needs to share its transmit line with other transmitters.

The EIA-485 interface uses the same channel to transmit and receive so the host must be able to switch its driver direction from transmit to receive. EIA-485 is robust but it is *very* unforgiving of timing and protocol mismatches between the host and the terminal. Be very careful in the design of your network interface and software and you will be rewarded with reliable communications; be complacent and you will regret it.

4.4.1.1 Turn-Around Time

The network driver on a multi-drop terminal must be switched on when the terminal is transmitting a packet, and switched off (high impedance) at all other times, so that other terminals can use the network. This is done automatically by the QTERM software. The host must control its own transmit line switching.

Turn-around time only has meaning on an EIA-485 multidrop network. Because a multidrop EIA-422 network uses separate channels to transmit and receive, the host need not switch its transmitter direction or implement a turn-around time delay before transmitting. Turn-around time is critical on an EIA-485 network.

On an EIA-485 network, after the host has sent the stop bit of the last byte in a packet to a network QTERM, it must switch its drivers to receive in less time than the QTERM's current turn-around time setting or the network will fail.

4.4.1.2 Network Biasing and Matching Resistors

On any multidrop network, there will be periods of inactivity when the network will not be actively driven by either the host or a QTERM. During these periods, the network may “float.” You must ensure that the differential voltage

between the + and - network lines cannot float enough to produce a phantom start bit at the QTERM or the QTERM will think it is receiving data when it is not. This will cause network communications faults and other problems which can be difficult to diagnose.

To prevent these problems, you must bias the network towards the “marking” state so that it will always enter the “marking” state when it is not being actively driven. Since both the EIA-422 and -485 networks are non-inverted, the simplest way to implement the needed bias is to “pull up” the “+” line by connecting a 1kΩ to 100kΩ resistor between the “+” line and power, and “pull down” the “-” line by connecting a similar resistor between the “-” line and ground. This will ensure that the network is always “marking” when idle. On an EIA-422, multidrop network, the QTERM's receive lines (“+” and “-”) are always driven by the host so you don't need to pull these lines up or down. The QTERM's transmit lines are sometimes tri-stated so these must be biased to prevent the host from seeing phantom noise at its receiver. The “marking” bias usually only needs to be applied at one point on the network.

If your network is very long (over 1000 feet or so) you may need to “terminate” the network; most networks will not require termination. This means installing an impedance matching resistor between the “+” and “-” lines on your network. On an EIA-422 network, you must “terminate” the transmit and the receive pairs separately, if you need to terminate the network at all. A typical matching-resistor value would be 147Ω, 1/4 watt. This resistor should be installed in the QTERM farthest from the host.

4.4.2 Multidrop Data Packets

All multidrop packets are formatted as follows:

<ADR1> <ADR2> <DATA...> <EOP>

where:

<ADR1>=Unit address MSD, ASCII decimal digit, 0 to 9
<ADR2>=Unit address LSD, ASCII decimal digit, 0 to 9
<DATA....>=Data bytes, zero or more data bytes
<EOP>=End-Of-Packet Terminator byte

Unit Address. Each QTERM attached to a multidrop network must have a unique address from “01” to “99”. This address can be set either during power-on setup or by a QDATA download. To send a packet to a QTERM, the host puts the unit's address in the ADR1 and ADR2 fields in the data packet. The response packet returned by the QTERM will *also* contain that unit's address in these fields.

Address “00” is used to indicate a *broadcast* packet. Broadcast packets are received and processed by all QTERMs connected to the network, regardless of their addresses. Since all the terminals on a network process broadcast packets at the same time, the QTERM *never* transmits a response packet in answer to a broadcast packet.

Data Bytes. The <DATA....> portion of the packet consists of zero or more bytes of data (i.e. an empty packet is valid). In packets sent by the host, the DATA portion will usually consist of commands and/or display data. In packets returned by the QTERM-, the DATA portion will always contain one of the following:

- an ASCII ACK or NAK for acknowledgment (byte values x06 and x15, respectively)
- a status byte
- a command response string
- the contents of the QTERM's EDIT buffer
- nothing (Null packet)

The only invalid value for a data byte (whether to or from the host) is the EOP Terminator byte.

End-of-Packet. The EOP character is used to terminate all data packets.

4.4.3 Receiving Data from the Host

Data received from the host is processed the same as in Block Mode.

4.4.4 Response Packets

Response packets are generated similarly to Block Mode. All response packets are prefixed with the unit address of the responding terminal. ACK and NAK response packets can be disabled with the QDATA file's **[MD/Block response mode]** field.

4.4.5 Transmitting Data to the Host and the Operation of EDIT Mode

This is also similar to Block Mode with one major difference: the EDIT Buffer *is not* sent when the QTERM encounters the EOP Terminator, within a string definition, while in EDIT mode. Instead the QTERM enters PENDING mode,

the LCD flashes and the keypad is disabled. *The LCD flash rate is controlled by the “Set LED Blink Rate” command.* If the flash rate is set too fast, the display will be difficult to read.

Once the QTERM enters PENDING mode, it will remain there until the host sends a packet containing the Query Buffer command. *The host must send a packet containing the "Query Buffer" command to receive the contents of the QTERM's EDIT buffer*. When the QTERM receives the "Query Buffer" command, its EDIT buffer will be transmitted to the host, PENDING mode will terminate, the LCD will stop flashing and the QTERM will return to IDLE mode to await the next event.

Data packets transmitted by a QTERM are always prefixed with the unit address of the responding QTERM.

CHAPTER 5.

QTERM-IV HARDWARE

5.1 Handheld QTERM-IV

The dimensions of the handheld version of the QTERM-IV are shown in Figure 4-1. This figure also shows the standard keypad legend for the bottom five rows of the 24-key keypad (the top row is labeled to user-specifications).

The housing is made from impact-resistant ABS and is colored black. The housing is not waterproof, but it can be subjected to moderate rain or splash without harm.

The handheld QTERM-IV uses either a 6-pin Modular or an optional DB9 female for connection to the external cable. The unit can be ordered with either connector exiting from the top or the bottom of the housing. If desired, the 6-pin Modular connector can easily be switched from one end of the housing to the other.

The pin assignments for the connectors for each of the available interfaces are shown in Table 5-1 (The lines *dig0* and *dig1* are discussed in section 5.5.3.). Figure 5-3 shows the pin numbering for each of the connectors used in the QTERM-IV. For reference, Table 1-5 (at the end of this chapter) shows the pin assignments used by the COM ports on PC-style computers for both DB25 and DB9 connectors.

5.2 Battery Back QTERM-IV

When ordered with the battery back, the handheld QTERM-IV does not have any power connection, i.e. the +V pin on the external connector is not used.

The battery back uses six “AA” size alkaline batteries. Typical battery life is shown in Table 5-3. Dimensions of the handheld with battery back are shown in Figure 4-1.

The battery back includes a power-supply board with an on/off switch. This board generates 5 volts DC from the batter-

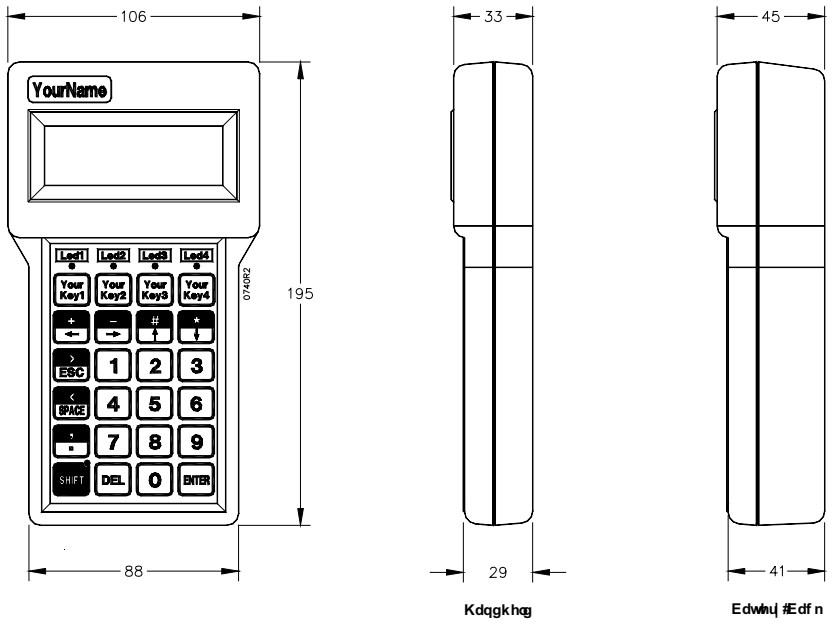


Figure 4-1. Dimensions of the Handheld and Battery-Back QTERM-IV.

ies, so the regulator option should *not* be ordered with the battery back option.

Because of the battery compartment, the connectors on a battery-back unit can only exit from the top of the case.

5.3 Panel-Mount QTERM-IV

The panel-mount QTERM-IV is mounted directly onto your instrument or enclosure. Figure 5-2 shows the dimensions of the panel-mount QTERM-IV as well as the mounting cutout required to install it.

The QTERM-IV is mounted to the panel via two steel brackets that span across the back of the panel cutout and are attached by four self-tapping screws that insert from the rear of the panel and screw into bosses on the QTERM-IV panel-mount enclosure. Screws and brackets are included with the QTERM-IV; these are usable with panels up to about 4 mm thick. If you are using a thicker panel, use M3 (#4) self-tapping screws that are about 8 mm longer than the panel thickness.

Connection to the panel-mount QTERM-IV is made via a 2-mm connector on the internal PC board which is accessible from the back of the mounting panel. The pin assignments for the panel-mount QTERM-IV connector are shown in Table 5-1; the lines *dig0* and *dig1* are discussed in section 5.5.3. The connector pin numbering is shown in Figure 5-3.

5.4 Interfaces

5.4.1 EIA-232 Interface

With proper cables and good grounding, the EIA-232 QTERM-IV can communicate up to about 15 meters at its top speed of 9600 baud. The QTERM-IV does not monitor or control any of the EIA-232 modem-control lines (such as RTS and DTS). Handshaking between the host and the QTERM-IV is done using software XON/XOFF commands. The XON/XOFF operation can be disabled (via software commands or the QDATA file) if you do not wish to have it operate.

5.4.2 EIA-422 Interface

With proper cables and grounding, the EIA-422 interface can communicate up to a distance of about 1000 meters. Since the EIA-422 version of the QTERM-IV requires four communication lines and two power lines, the *dig0* and *dig1* lines are not available.

The EIA-422 unit does not have the 120Ω line termination resistor often used to terminate an EIA-422 interface. If you need this resistor, you can install it on the main PC board (for handheld units, remove the back of the case). Figure 5-4 shows where the resistor is to be soldered in. Do *not* attempt to remove the PC board from the case!

For volume orders, Beijer Electronics can supply the QTERM-IV with this resistor installed.

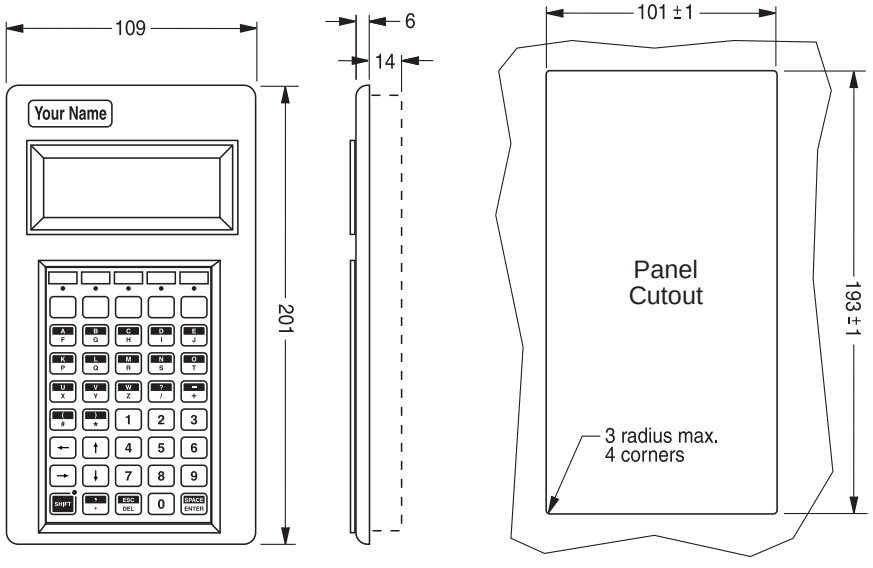


Figure 5-2. Dimensions of the Panel-Mount QTERM-IV.

Table 5-1: QTERM-IV Pin Assignments.

Handheld 6-pin Modular	Handheld DB9	QTERM-IV Panel Mount 2 mm	QTERM-IV 232/5 Volt Buffered Function	QTERM-IV 422 Function	QTERM-IV 485 Function
1	3	1	receive	+receive	dig0*
2	6	2	dig1*	-receive	dig1*
3	2	3	transmit	+transmit	RTx+
4	1	4	dig0*	-transmit	RTx-
5	9	5	+V†	+V†	+V†
6	5	6	ground	ground	ground

Tx and *Rx* directions are with respect to the QTERM-IV
*these lines are optional; see section 5.5.3
†no connection with battery back units

5.4.3 5-volt Buffered Interface

When used to communicate with another 5-volt Buffered device, the QTERM-IV can operate at distances up to 5 meters. The advantage of the 5-volt Buffered QTERM-IV is that it is lower power than the EIA-232 version and uses the same 5-volt logic levels as come directly out of a UART (i.e. the idle state is at 5 volts, and the active state is at 0 volts).

Although the signal levels are the same as at a UART, the transmit and receive lines are buffered and can be connected to live signals whether or not the QTERM-IV/P40 is powered (turned on).

5.5 Other QTERM-IV Hardware

5.5.1 Display

The QTERM-IV display is a 4-line by 20-character supertwist unit. The ASCII character set and eight user-defined characters can be displayed. Hosts which transmit 8-bit data can also display an additional 64 characters including Greek letters, katakana characters, non-English alphabetic characters and math symbols.

Appendix B is a chart which shows what the QTERM-IV does with every possible 8-bit value it can receive. Note that the ASCII portion of the chart (the first 128 characters) is similar, *but not identical*, to the true ASCII chart shown in Appendix A.

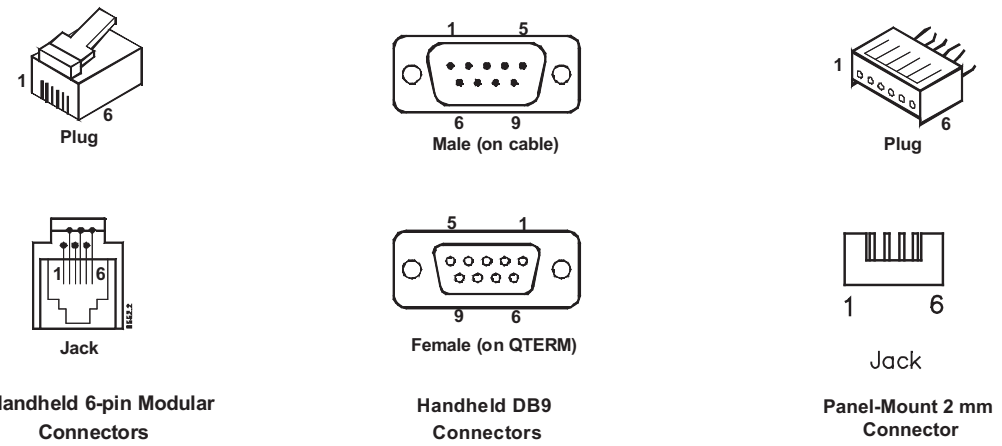


Figure 5-3. QTERM-IV Connector Pin Numbering.

If you ordered the backlight, the software commands shown in Chapter 2 allow you to turn the backlight on and off. Without the backlight, these commands have no effect.

If you have a wide-temperature display (which includes a backlight), it will operate identically to the backlit display.

5.5.2 Keypad

The lower-left key on both the 24- and 40-key keypads is a shift key and has an associated LED to indicate the shift function (this can be changed by the QDATA file; see Chapter 3). Above the top row of keys is a row of user-programmable LEDs (four on the 24-key keypad, and five on the 40-key keypad).

Software commands and the QDATA file allow you to control both key clicks (on or off) and key repeat (on or off, independently programmable for each key).

Every key can be programmed to return an unshifted and a shifted character or string (up to 255 characters, including delays, subject to the limitation noted in Chapter 3). Every key can also be programmed to return an unshifted and shifted release character or string.

5.5.3 Digital Outputs, *dig0* and *dig1*

The QTERM-IV has two programmable digital output bits (*dig0* and *dig1*) which can be used to control external devices. (Note that these are not available on the EIA-422 version due to lack of connector pins.)

Software commands can be used to set these lines high or low, or to have one or both function as an external buzzer or horn signal. Both lines are 74HC bus-driver outputs, and can sink or source up to about 20 mA of current at 0 or 5 volts, respectively.

If either *dig0* or *dig1* is programmed to act as an external buzzer signal, it acts as an enable signal: when the line is low (0 volts), it should shut off the external buzzer or horn; when the line is high (5 volts), it should turn on the external buzzer or horn. These lines should only be used to drive an external transistor which in turn drives the external buzzer.

5.5.4 Buzzer

The QTERM-IV includes an audio buzzer, which is used for key clicks, for beeping in response to a "bell" character (^G, 07h) and for programmable-duration buzzing.

For applications in environments that are too loud for the buzzer to be heard, the digital outputs (described above) can be programmed to trigger an external device when buzzer commands are received. (These signals are not available on EIA-422 units due to lack of connector pins.)

5.5.5 Regulator Option

The standard QTERM-IV requires a 4.8- to 5.5-volt regulated, SELV, DC power source. An optional regulator allows the QTERM-IV to be operated from a 5.5- to 24-volt SELV, DC source. The power supply should have a current limit on its output of 5 Amperes. Limiting may be inherent to the supply or may be provided by supplementary devices.

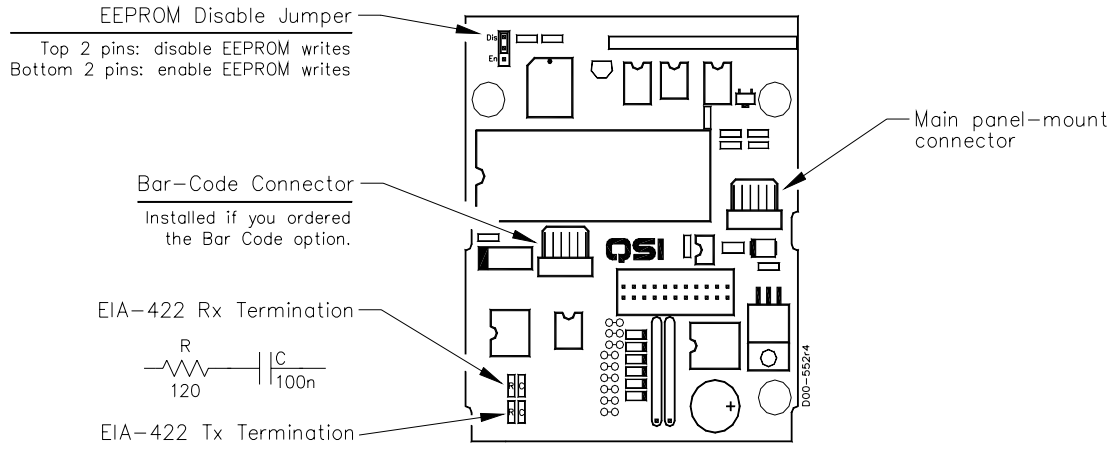


Figure 5-4. Installing the EIA-422 Termination Resistor; Locating the Bar Code Connector.

5.5.6 Auxiliary Serial Port Option

If you ordered the Auxiliary Port option with your QTERM-IV, you have an additional connector. On handheld units, this is a DB9 connector which exits from the same end of the case as the 6-pin Modular connector; on panel-mount units, this is a 6-pin, 2 mm connector (Figure 5-4).

Table 1-2 shows the pin assignments for both the panel-mount and handheld connectors. The handheld DB9 will mate directly with the DB9 male on an HP SmartWand bar code reader.

The auxiliary serial port operates at 1200 baud with a data format of 8n1. The QTERM-IV host port must operate at 1200 baud or faster when using the auxiliary port option. This serial port does not support XON/XOFF handshaking. See section 2.2.46 for a detailed description of the operation of the auxiliary port.

5.5.7 The EEPROM Disable Jumper

Figure 5-4 shows the location of the EEPROM disable/enable jumper. This is shipped in the *enable* position, which allows the EEPROM to be modified with QDATA downloads or the *Save Parameters to EEPROM* command.

If you do not want the EEPROM to be modified under any conditions, move this jumper from the *enable* to the *disable* position. This will prevent any writes to the EEPROM, including downloading of QCODE or QDATA files.

Table 1-2: Auxiliary Port Connector Pin Assignments.

Handheld DB9	Panel-Mount 2 mm	Function
2	1	receive from device
3	3	transmit to device
9	5	5 volts DC to device
7	6	ground

5.5.8 Adding Another PC Board

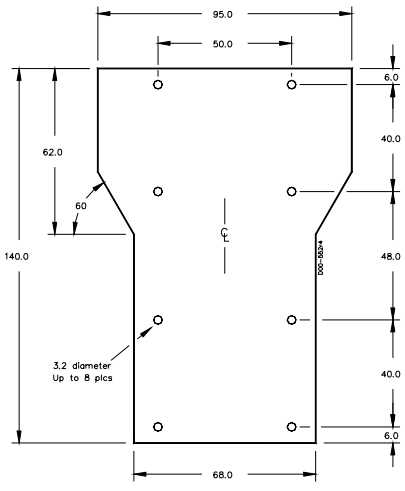
If you are using a handheld or battery back QTERM-IV, you can install an additional PC board in the back of the case. This board can be used, of course, for whatever purpose your application requires.

Figure 5-5 shows the physical size of this printed circuit board, as well as the allowable component height. There are plastic bosses molded into the back of the case; you can use all or just some of these for mounting your PC board. The ones you do not wish to use can be easily clipped off.

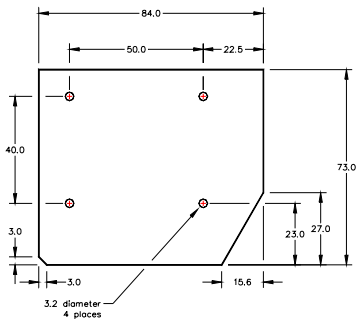
For the bosses that you do want to use, place a 3.2 mm diameter hole in your PC board. After the PC board is installed over the bosses, you can either press on a retainer snap ring (Eaton-Tinnerman part number T99220SS-67-01-10780) or heat-stake the bosses.

5.6 QTERM-IV Specifications

Table 5-3 lists various specifications for the QTERM-IV. Note that all components of the QTERM-IV, other than the



For Handheld Unit



For Battery-Back Unit

Figure 5-5. Size of Additional PC Board (4).

Table 1-3: Environmental and Power Specifications.

Parameter	Limits
Operating Temperature Range (except for display)	-40 to 85°C
Standard/Backlit Display usable temperature range	-10 to 60°C
Wide-temperature Display usable temperature range	-20 to 70°C
Storage temperature, all components	-40 to 85°C
Maximum humidity range, all components	0 to 95%, non-condensing
Operating voltage range, without regulator option	4.8 to 5.5 volts
Operating voltage range, with regulator option	5.5 to 24 volts
Battery Back Option	
Battery life, EIA-232	150 hours
Battery life, EIA-422	190 hours
Battery life, 5-volt Buffered	190 hours
Battery life, backlight on	43 hours

display, will operate over the industrial temperature range of -40 to 85 °C. Table 1-4 lists current consumption of the various versions of the QTERM-IV.

Table 1-4: QTERM Current Consumption (mA).

Version	Ireg Active	Ireg Disp Off
EIA-232 version	20	12
5-volt Buffered version	17	9
EIA-422 version	17	9
Add for backlight (when on)	60	-
Add for wide-temp	2	0
Add for each LED on keypad (when on)	4	4
Add for regulator	1	1
Add for bar code wand (w/o wand)	3	3

Table 1-5. Pin Assignments for PC-Style COM Ports.

COM Ports with DB25 Male Connectors		COM Ports with DB9 Male Connectors	
Pin	**Function**	**Pin**	**Function**
2	PC transmit	1	CD *
3	PC receive	2	PC receive
5	CTS *	3	PC transmit
6	DSR *	5	ground
7	ground	6	DSR *
8	CD *	8	CTS *
22	RI *	9	RI *

*These lines normally can be left unconnected. Some PCs may require that one or more of them be pulled to 5 volts through a pullup resistor (about 300 W)

CHAPTER 6.

QTERM-P40 HARDWARE

6.1 Panel-Mount QTERM-P40

The dimensions of the QTERM-P40 are shown in Figure 6-1. This terminal is available with a 24-key or a 40-key keypad each with user-specified LED labels and function keys.

The housing is made from rugged glass-fiber filled nylon plastic and is colored black. The housing is not waterproof, but it can be subjected to moderate rain or splash without harm.

The QTERM-P40 is mounted directly onto your instrument or enclosure panel. Figure 4-1 shows the mounting cutout required to install the QTERM-P40.

The QTERM-P40 is mounted to the panel via four angle brackets that attach to the terminal with self-tapping screws that insert from the rear of the bracket and screw into bosses on the QTERM-P40 enclosure. These brackets are designed so they can be attached before inserting the terminal into the panel. Once the brackets have been installed, the terminal is inserted into the panel and the brackets are slid outward. Then the screws are tightened and the brackets hold the terminal in place. Screws and brackets are included with the QTERM-P40; these are usable with panels up to about 10 mm thick. A gasket is provided to seal the outside edges of the terminal where it presses against the panel. Connection to the panel-mount QTERM-P40 is made via the ten-pin terminal strip or the DB9 connector which is accessible on the back of the terminal. The lines *dig0* and *dig1* are discussed in section 6.3.3. The connector pin numbering is shown in Figure 6-2.

6.2 Interfaces

6.2.1 EIA-232 Interface

With proper cables and good grounding, the EIA-232 QTERM-P40 can communicate up to 15 meters at its top speed of 9600 baud.

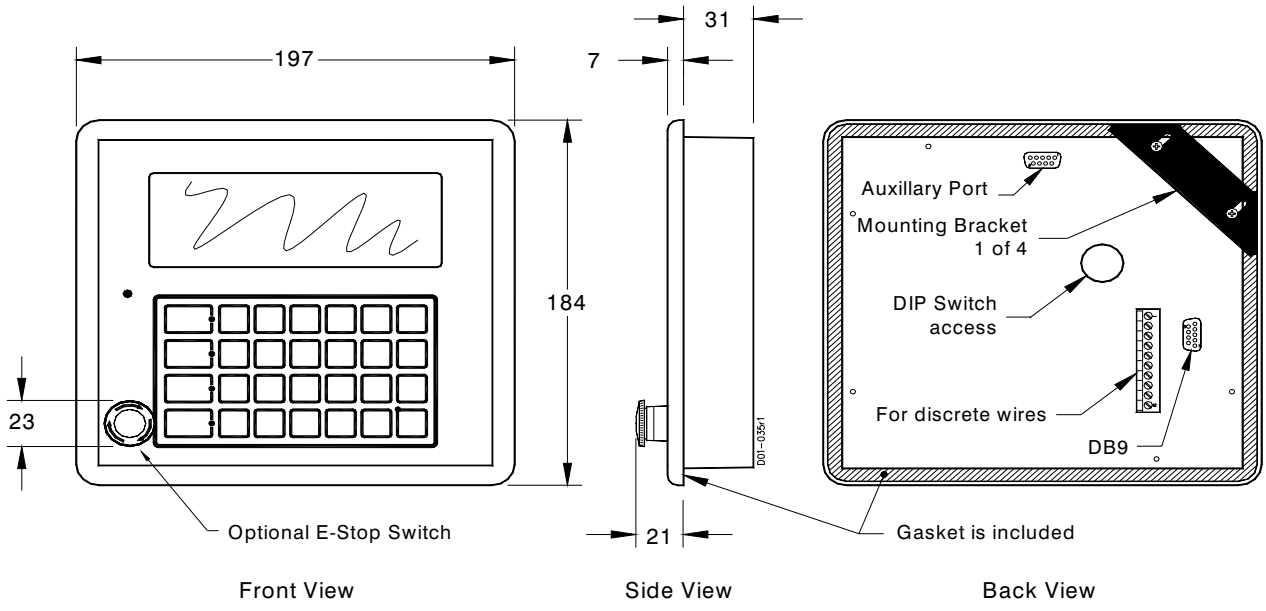


Figure 6-1. Dimensions of the QTERM-P40 Panel-Mount Terminal.

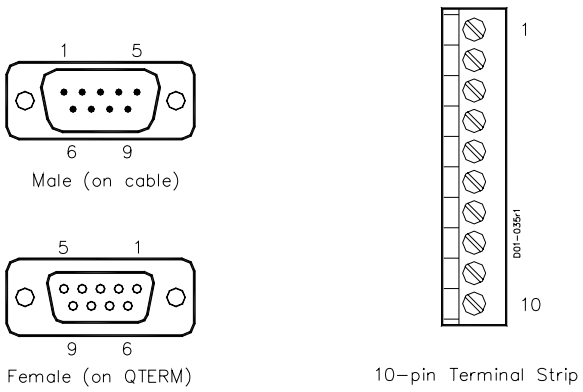


Figure 6-2. QTERM-P40 Connector Pin Numbering.

The QTERM-P40 does not monitor or control any of the EIA-232 modem-control lines (such as RTS and DTS). Handshaking between the host and the QTERM-P40 is done using software XON/XOFF commands. The XON/XOFF operation can be disabled (via software commands or the QDATA file) if you do not wish to have it operate.

6.2.2 EIA-422 Interface

With proper cables and grounding, the EIA-422 interface can communicate up to a distance of 1000 meters. Since the EIA-422 version of the QTERM-P40 requires four communication lines and two power lines, the *dig0* and *dig1* lines are not available.

The EIA-422 unit does not have the 120Ω line termination resistor often used to terminate an EIA-422 interface. If you need this resistor, you can install it on the main PC board. Figure 6-4 shows where the resistor is to be soldered in. **Do *not* attempt to remove the PC board from the case.**

For volume orders, Beijer Electronics can supply the QTERM-IV with this resistor installed.

6.2.3 5-volt Buffered Interface

When used to communicate with another 5-volt Buffered device, the QTERM-P40 can operate at distances up to 5 meters. The advantage of the 5-volt Buffered QTERM- P40 is that it is lower power than the EIA-232 version, and uses the same 5-volt logic levels as come directly out of a UART (i.e. the idle state is at 5 volts, and the active state is a 0).

Table 6-1: QTERM-P40 Pin Assignments.

DB9 Female	10-Pin Terminal Strip	QTERM-P40 232/5-volt Buffered Function	QTERM-P40 422 Function	QTERM-P40 485 Function
3	4	receive	+receive	dig0*
6	5	dig1*	-receive	dig1*
2	6	transmit	+transmit	RTx+
1	7	dig0*	-transmit	RTx-
9	8	+V	+V	+V
5	9	ground	ground	ground
n/a	1	E-stop N.O.	E-stop N.O.	E-stop N.O.
8	2	E-stop com.	E-stop com.	E-stop com.
7	3	E-stop N.C.	E-stop N.C.	E-stop N.C.

Tx and *Rx* directions are with respect to the QTERM-P40
*these lines are optional; see section 6.3.3

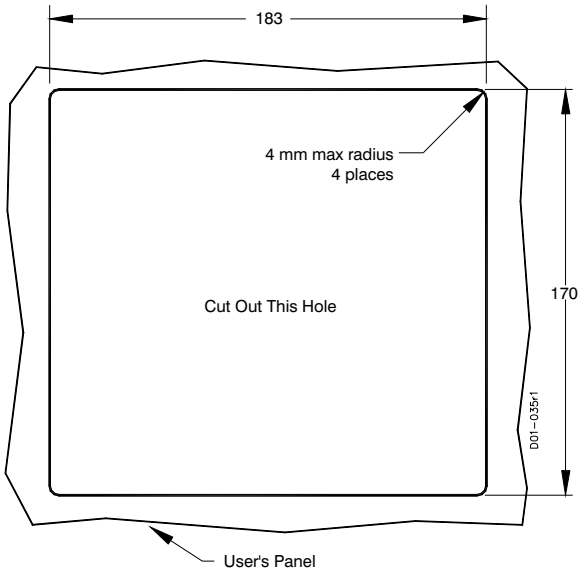


Figure 4-1. Cutout Dimensions for the QTERM-P40.

Although the signal lines are the same as at a UART, the transmit and receive lines are buffered and can be connected to live signals whether or not the QTERM-P40 is powered (turned on).

6.3 Other QTERM-P40 Hardware

6.3.1 Display

The QTERM-P40 display is a 4-line by 20-character supertwist unit with an LED backlight and large (8.1mm) characters. The ASCII character set and eight user-defined characters can be displayed. Hosts which transmit 8-bit data can also display an additional 64 characters including Greek letters, katakana characters, non-English alphabetic characters and math symbols.

Appendix B is a chart which shows what the QTERM-P40 does with every possible 8-bit value it can receive. Note that the ASCII portion of the chart (the first 128 characters) is similar, *but not identical*, to the true ASCII chart shown in Appendix A.

6.3.2 Keypad

The lower right key on both the 24- and 40-key keypads is a shift key and has an associated LED to indicate the shift function (this can by changed by the QDATA file; see Chapter 3). Along the left side is a column of user-programmable LEDs (four on the 24-key keypad and five on the 40-key keypad).

Software commands and the QDATA file allow you to control both key clicks (on or off) and key repeat (on or off, independently programmable for each key).

Every key can be programmed to return an unshifted and a shifted character or string (up to 255 characters, including delays, subject to the limitation noted in Chapter 3). Every

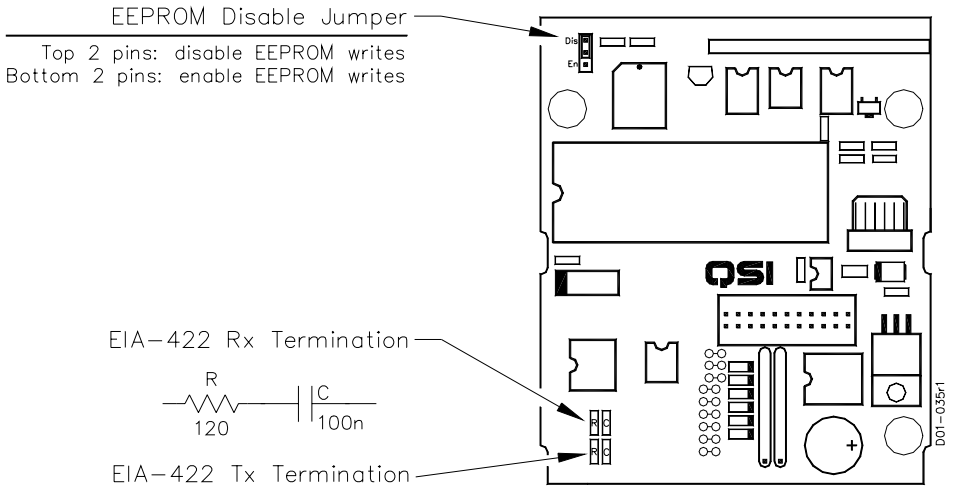


Figure 6-4. Installing the EIA-422 Termination Resistor; Locating the EEPROM Disable Jumper.

key can also be programmed to return an unshifted and shifted release character or string.

6.3.3 Digital Outputs, *dig0* and *dig1*

The QTERM-P40 has two programmable digital output bits (*dig0* and *dig1*) which can be used to control external devices. (Note that these are not available on the EIA-422 version due to lack of connector pins.)

Software commands can be used to set these lines high or low, or to have one or both function as an external buzzer or horn signal. Both lines are 74HC bus-driver outputs and can sink or source up to about 20 mA of current at 0 or 5 volts, respectively.

If either *dig0* or *dig1* is programmed to act as an external buzzer signal, it acts as an enable signal: when the line is low (0 volts), it should shut off the external buzzer or horn; when the line is high (5 volts), it should turn on the external buzzer or horn. These lines should only be used to drive an external transistor which in turn drives the external buzzer.

6.3.4 Buzzer

The QTERM-P40 includes an audio buzzer, which is used for key clicks, for beeping in response to a "bell" character (^G, 07h) and for programmable-duration buzzing.

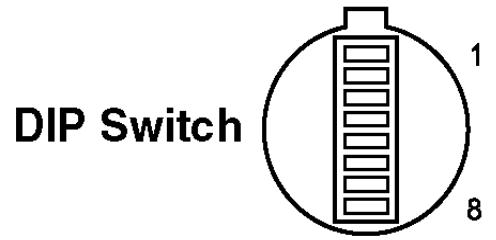
For applications in environments that are too loud for the buzzer to be heard, the digital outputs (described above) can be programmed to trigger an external device when buzzer commands are received. (These signals are not available on EIA-422 units due to lack of connector pins.)

6.3.5 Regulator Operation

The QTERM-P40 has a built-in 5 volt regulator. This allows the QTERM-P40 to be operated from a 5.5- to a 24-volt SELV, DC source. The power supply should have a current limit on its output of 5 Amperes. Limiting may be inherent to the supply or may be provided by supplementary devices. If you have a regulated 5 volt source to power the terminal, then you can disable the regulator using the dip-switches. Dip switch settings are shown in Figure 6-5.

Table 6-2: Aux Port Connector Pin Assignments.

Handheld DB9	Panel-Mount 2 mm	Function
2	1	receive
3	3	transmit
9	5	5 volts DC
7	6	ground



DIP Switch Settings

	EIA-232 EIA-423	EIA-422 EIA-422M	EIA-485
1	Open if Pwr=5 VDC	Closed if Pwr=7.5-24 VDC	
2	Closed	Closed	Connects DIG0
3	Connects DIG1	Closed	Connects DIG1
4	Connects DIG0	Closed	Closed
5	Closed if Pwr=5 VDC	Open if Pwr=7.5-24 VDC	
6	Chassis GND to Signal GND Closed/Open		
7	422Tx/485 Termination Closed/Open		
8	422Rx Termination Closed/Open		

Figure 6-5. Dip Switch Settings for the QTERM-P40

6.3.6 Auxiliary Serial Port Option

If you ordered the auxiliary port option with your QTERM-P40, you have an additional connector. This is a DB9 connector, which exits from the top of the rear of the case, and is labeled AUXILIARY PORT (see Figure 6-1).

Table 6-2 shows the pin assignments for both the panel-mount and handheld connectors. The handheld DB9 will mate directly with the DB9 male on an HP SmartWand bar code reader.

The auxiliary serial port operates at 1200 baud with a data format of 8n1. The QTERM-P40 host port must operate at 1200 baud or faster when using the aux port option. This serial port does not support XON/XOFF handshaking.

6.3.7 The EEPROM Disable Jumper

Figure 6-4 shows the location of the EEPROM disable/enable jumper. This is shipped in the *enable* position, which allows the EEPROM to be modified with QDATA downloads or the *Save Parameters to EEPROM* command.

If you do not want the EEPROM to be modified under any conditions, move this jumper from the *enable* to the *disable* position. This will prevent any writes to the EEPROM, including downloading of QCODE or QDATA files.

6.3.8 The Emergency Stop Switch

The QTERM-P40 has an optional Emergency Stop Switch (E-stop) that can be used for triggering some external event or device. This switch is a push-once, twist to reset, mushroom shaped switch mounted on the front lower left side of the terminal.

Connection to the E-stop switch is made through the interface connectors on the back of the QTERM-P40. The DB9 connector has leads for the common and normally closed switch contacts. The 10-pin terminal strip has leads for the common, normally closed and normally open switch contacts (see Table 6-1 for pin designations). This switch is rated for a maximum of 1 amp at 30VDC.

6.4 QTERM-P40 Specifications

Table 6-3 lists various specifications for the QTERM-P40. Note that all components of the QTERM-P40, other than the display, will operate over the industrial temperature range of -40 to 85 °C. Table 6-4 lists current consumption of the various versions of the QTERM-P40.

Table 6-3: Environmental and Power Specifications.

Parameter	Limits
Operating Temperature Range (except for display)	-40 to 85°C
Standard/Backlit Display usable temperature range	-10 to 60°C
Storage temperature, all components	-40 to 85°C
Maximum humidity range, all components	0 to 95%, non-condensing
Operating voltage range, with regulator disabled	4.8 to 5.5 volts
Operating voltage range, with regulator enabled	5.5 to 24 volts

Table 6-4: QTERM-P40 Current Consumption (mA).

Version	Ireg Active	Ireg Disp Off
EIA-232 version	25	15
5-volt Buffered version	22	12
EIA-422 version	22	12
Add for backlight (when on)	150	-
Add for each LED on keypad (when on)	4	4
Add for regulator	1	1
Add for bar code wand (w/o wand)	3	3

APPENDIX A.

DVFLI#F KDUW

Least Significant Digit (hex)	Most Significant Digit (hex)							
	3	4	5	6	7	8	9	:
3	QXO	GOH	VS	3	C	S	c	s
4	VRK	GF4	$	4	D	T	d	t
5	VW[	GF5	%	5	E	U	e	u
6	HW[	GF6	&	6	F	V	f	v
7	HRW	GF7	’	7	G	W	g	w
8	HQT	QDN	(	8	H	X	h	x
9	DFN	V\ Q	)	9	I	Y	i	y
:	EHO	HWE	*	:	J	Z	j	z
;	EV	FDQ	+	;	K	[	k	{
<	KW	HP	,	<	L	\	l	|
D	OI	VXE	-	=	M	]	m	}
E	YW	HVF	.	>	N	^	n	~
F	I I	I V	/	?	O	_	o	¡
G	FU	J V	0	@	P	‘	p	¢
H	VR	UV	1	A	Q	a	q	£
I	VL	XV	2	B	R	b	r	GHO

0345R3

NUL = blank
SOH = start of header
STX = start of text
ETX = end of text
EOT = end of transmission
ENQ = enquiry
ACK = acknowledge
BEL = bell
BS = backspace
HT = horizontal tab
LF = line feed
VT = vertical tab
FF = form feed
CR = carriage return
SO = shift out
SI = shift in
SP = space
DLE = data link escape
DC1 = device control 1 (XON)
DC2 = device control 2
DC3 = device control 3 (XOFF)
DC4 = device control 4
NAK = negative acknowledge
SYN = synchronization
ETB = end of text block
CAN = cancel
EM = end of medium
SUB = substitute
ESC = escape
FS = file separator
GS = group separator
RS = record separator
US = unit separator
DEL = delete/rubout

APPENDIX B.

QTERM CHARACTER CHART

The chart on the next page shows how the QTERM responds to each of the 256 possible values of characters which it can receive.

Where a dot pattern is shown, sending the corresponding code will cause the QTERM to display the dot pattern at the current cursor location.

Numbers in circles refer to these notes:

❶ These bytes are always ignored. They cannot be used for remapping custom characters.
❷ These bytes are normally ignored but may be used for remapping custom characters if desired.
❸ This is a space character.

Custom characters, and the codes to which they are assigned, are defined in the QDATA file (see Chapter 3). The following restrictions apply to assigning the codes:

- Custom characters may *not* be mapped to any code from 07h through 1Bh (inclusive). This includes the ❶ bytes as well as the control codes which are used by the QTERM.
- Custom characters *may* be mapped to any other code, i.e. 00h through 06h and 1Ch through FFh.
- The characters shown in the chart from 00h through 06h and at 1Ch are the default custom character definitions. You may change the character dot patterns and map them to different codes using the QDATA setup file.

Other notations in the chart are:

BEL	bell (beep) command
BS	backspace
HT	horizontal tab
LF	line feed
CR	carriage return
XON	XON character
XOFF	XOFF character
ESC	escape character
DEL	delete character

Note that, although the left half of this chart is similar to the ASCII chart in Appendix A, there are differences.

Least Significant Digit (hex)	Most Significant Digit (hex)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		①	③						②	②	③					
1		XON							②	②						
2		①							②	②						
3		XOFF							②	②						
4		①							②	②						
5		①							②	②						
6		①							②	②						
7	BEL	①							②	②						
8	BS	①							②	②						
9	HT	①							②	②						
A	LF	①							②	②						
B	LF	ESC							②	②						
C	LF								②	③						
D	CR	③							②	③						
E	①	③							②	③						③
F	①	③						DEL	②	③						

0553R3a

(Notes ❶, ❷ and ❸ are on the previous page.)

APPENDIX C.

QTERM COMMAND SUMMARY

This appendix is an abbreviated summary of all of the available QTERM software commands. More detailed descriptions of the commands are in Chapter 2.

Refer to Table 1-2 for a complete list of acceptable parameter values for applicable commands. If a command parameter is out of the valid range, the QTERM ignores the entire command.

Note that the timing shown for the execution of the various commands is only approximate. Many factors can affect the execution time, so it is impossible to give exact figures.

Command	String	Timing (ms)	Notes & Parameters
display character		0.7 14.0 37.0	typical in replace mode typical in insert mode worst case in insert mode
Bell (^G)	07H	0.6	other buzzer commands not executed until beep is over
Backspace (^H)	08H	0.8	
Horizontal Tab (^I)	09H	0.8	1.1 ms worst case
Line Feed (^J)	0AH	0.8	up to 35 ms if auto scroll is on
Vertical Tab (^K)	0BH		same as line feed
Form Feed (^L)	0CH		same as line feed
Carriage Return (^M)	0DH	0.8	up to 35 ms if auto scroll is on
XON (^Q)	11H	0.4	
XOFF (^S)	13H	0.4	
Delete	7FH	1.2	
Cursor Up	⍋ A	0.8	
Cursor Down	⍋ B	0.8	
Cursor Right	⍋ C	0.8	
Cursor Left	⍋ D	0.8	
Clear Screen	⍋ E	4.4	
Set Timeout Delay	⍋ F#	1.2	see Table 1-2
Set Tab Spacing	⍋ G#	1.2	#= @ causes tabs to be ignored A to S for tab set every 1 to 19 spaces
Cursor Home	⍋ H	3.7	

Command	String	Timing (ms)	Notes & Parameters
Set Cursor Position	ESC I#*	1.5	#= @ to C for row 0 to 3 *= @ to S for column 0 to 19
Erase to End of Screen	ESC J	6.0 12.0 18.0 24.0	if cursor is on row 3 if cursor is on row 2 if cursor is on row 1 if cursor is on row 0
Erase to End of Line	ESC K	3.0 5.8	typical maximum (cursor in column 0)
Set Contrast	ESC L#	1.2	#= @ to DEL for lightest to darkest
Reset QTERM	ESC M	300.0	
Query Version	ESC N	0.5	time to load characters into transmit buffer
Buzzer On/Off/Beep	ESC O#	1.2	other buzzer commands not executed until beep is over #= @ turn buzzer off = A turn buzzer on = B beep for 1/2 second
LED Control On/Off/Blink/Toggle	ESC P#	1.2	#= @ to U; see Table 1-2
Insert/Replace Mode	ESC Q#	1.2	#= @ set replace mode A set insert mode
Auto Wrap Mode	ESC R#	1.2	#= @ auto wrap off A auto wrap on
Auto Scroll Mode	ESC S#	1.2	#= @ auto scroll off A auto scroll on
Auto Line Feed Mode	ESC T#	1.2	#= @ auto line feed off A auto line feed on
Display Off	ESC U#	2.0	key press or character received turns display back on
Backlight Mode	ESC V#	1.2	#= @ backlight off A backlight on B backlight toggle
Query Status	ESC W	0.5	time to load characters into transmit buffer see Table 1-2 for returned values
Query Cursor Position	ESC X	0.5	time to load characters into transmit buffer see Table 1-2 for returned values
Query Character	ESC Y	0.5	time to load character into transmit buffer
Scroll Down	ESC Z	35.0	
Key Repeat/Click Mode	ESC a#	1.2	#= @ click off, repeat off #= A click off, repeat on #= B click on, repeat off #= C click on, repeat on

Command	String	Timing (ms)	Notes & Parameters
Set Cursor mode	Ɛ b#	1.2	#= @ block off, underline off (no cursor) A block off, underline on B block on, underline off C block on, underline on
Set Shift Mode	Ɛ c#	1.2	#= @ function mode, shift indicator enabled A lock mode, shift indicator enabled B function mode, shift indicator disabled C lock mode, shift indicator disabled
Control *dig1* and *dig2*	Ɛ d#	1.2	see section 2.2.40 for values
Set Buzzer Duration	Ɛ e#	1.2	see Table 1-2
Set Key Click Duration	Ɛ f#	1.2	see Table 1-2
Set Key Repeat Rate	Ɛ g#	1.2	see Table 1-2
Set LED Blink Rate	Ɛ h#	1.2	see Table 1-2
Save Parameters to EEPROM	Ɛ i	6.0	see note in section 2.2.45
Auxiliary Port Control	Ɛ j#..	1.2 1.2 *	#= @ disable receive from Auxiliary Port A enable receive from Auxiliary Port B send string, *= about 8 ms per character
Transmit Buffer Flush	Ɛ k	1.2	
XON/XOFF Mode	Ɛ l#	1.2	#= @ disable XON/XOFF operation A enable XON/XOFF operation
User Area Read/Write	Ɛ m#		timing depends on baud rate # = @ read user; returns byte count and data bytes; see section 2.2.49 for details A write user data; follow "A" with byte count and data bytes; see section 2.2.49 for details
Get Free User Area	Ɛ n	2.2	see Table 1-2
User Area Page Control	Ɛ o	1.2	
Macro String Execution	Ɛ p		timing depends on macro string
Query Multidrop Buffer	Ɛ q		timing depends on number of characters in buffer